Supply Usage Requirements Estimator (SURE) Version 2.0
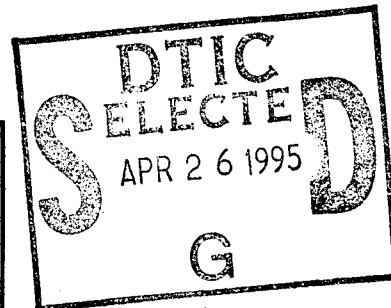
A Monograph
by

Major John M. Friedson

Quartermaster Corps

DTIC
SELECTED
APR 2 6 1995
G

MENS EST CLAVIS VICTORIAE

School of Advanced Military Studies
United States Army Command and General Staff College
Fort Leavenworth, Kansas

Second Term AY 93-94

19950425 112

REPORT DOCUMENTATION PAGE

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	6 MAY 1994	MONOGRAPH

4. TITLE AND SUBTITLE
SUPPLY USAGE REQUIREMENTS ESTIMATOR (SURE) VERSION 2.0 (U)

5. FUNDING NUMBERS

6. AUTHOR(S)

MAJ JOHN M. FRIEDSON, USA

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
SCHOOL OF ADVANCED MILITARY STUDIES
ATTN: ATZL-SWV
FORT LEAVENWORTH, KANSAS 66027-6900

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

SEE ATTACHED

14. SUBJECT TERMS

LOGISTICS
LOGISTICS PLANNING
MILITARY SUPPLIES
COMPUTER PROGRAMS

15. NUMBER OF PAGES
452

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UNLIMITED

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to *stay within the lines* to meet *optical scanning requirements*.

Block 1. Agency Use Only (*Leave blank*).

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C	- Contract	PR	- Project
G	- Grant	TA	- Task
PE	- Program Element	WU	- Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD	- See DoDD 5230.24, "Distribution Statements on Technical Documents."
DOE	- See authorities.
NASA	- See Handbook NHB 2200.2.
NTIS	- Leave blank.

Block 12b. Distribution Code.

DOD	- Leave blank.
DOE	- Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
NASA	- Leave blank.
NTIS	- Leave blank.

Block 13. Abstract. Include a brief (*Maximum 200 words*) factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code (*NTIS only*).

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

SCHOOL OF ADVANCED MILITARY STUDIES
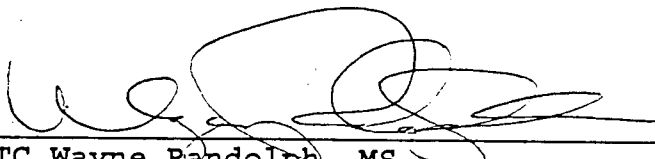
MONOGRAPH APPROVAL

Major John M. Friedson

Title of Monograph: Supply Usage Requirements Estimator (SURE)
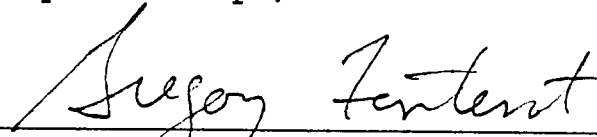
Version 2.0

Approved by:

_____ Monograph Director
LTC Wayne Randolph, MS

_____ Director, School of
COL Gregory Fontenot, MA, MMAS   Advanced Military
                                                    Studies

_____ Director, Graduate
Philip J. Brookes, Ph.D.   Degree Program

Accepted this 6th day of May 1994

ABSTRACT

SUPPLY USAGE REQUIREMENTS ESTIMATOR (SURE) VERSION 2.0 by
Major John M. Friedson, USA, 448 pages.

SURE 2.0 is a compiled, stand-alone dBase IV version 2.0 system that allows
planners to estimate detailed equipment-based consumption for supply classes III
and V, and population-based consumption for classes I, II, IV, VI, and VIII, and
water. It is a replacement for the previous SURE 1.x system that required a copy
of Lotus 1-2-3 version 2.x. Hardware requirements for SURE 2.0 are a DOS-based
system with 2MB memory or greater, and a hard drive with 3MB or more free disk
space available for installation. Use of SURE 2.0 may require from two to twenty
MB additional disk space depending on user entries.

SURE 2.0 uses doctrinal methods for estimating logistics requirements by
providing equipment data and formulas for a number of environmental and combat
conditions. SURE's concept of orders-based planning integrates phased operations,
parameter sets of operational conditions, and task-organized groupings of units to
facilitate the correlation of logistics estimates to operations plans and orders.

TABLE OF CONTENTS

CHAPTER I
INTRODUCTION

The supply system must be responsive, efficient, and simple to operate. It must be able to adjust to requirements of supported units and be able to forecast and satisfy fluctuating demands.[1] *FM 101-10-1/2*

Chapter 2 of Field Manual 101-10-1/2, *Staff Officers' Field Manual Organizational, Technical, and Logistical Data Planning Factors (Volume 2)* is the Army's primary handbook for estimating the food, water, clothing, equipment, arms, ammunition, fuel, and materials needed to conduct combat operations. It provides doctrinal methods for estimating those requirements through by providing equipment data and formulas for a number of environmental and combat conditions. The large number of possible Army units, wide variety of equipment, a and multiple permutations of environmental and combat conditions, however, makes manual calculation of these estimates difficult and time-consuming.

In an effort to overcome these limitations, the first commonly fielded program to automate the estimate process is the currently fielded Supply Usage Requirements Estimator, SURE 1.x.[2] This system uses a series of Lotus 123 version. 2.x templates that "allow planners to estimate detailed equipment-based consumption for supply classes III and V, and population-based consumption for classes I, II, IV, VI, and VIII, and water."[3] This system has the advantages of being able to be run on machines meeting a minimal hardware requirements, but its capabilities are limited by the programming constraints common to macro-based systems. As a result, SURE 1.0 has only limited capacity to save user work, is limited to calculations for a single day's consumption, and has no help system. Additionally, it requires the user to own a copy of Lotus Development Corporation's Lotus 1-2-3™, and a version 2.x copy in particular. To overcome these disadvantages, U.S. Army Combined Arms Support Command (CASCOM) Force Development and Evaluation Directorate (FD&E) decided that a new version of SURE, a database-based,

1

compiled (and thus a stand-alone program) version should be the next step in the SURE developmental process.

CASCOM's requirements for SURE 2.0, the compiled version, was for it to at least duplicate current SURE 1.0 capabilities, to be developed under one of a limited number of database systems, be dBase™ file compatible, and to offer as many as possible of a number of additional enhancements. Possible enhancements included, but were not limited to such features as a user-friendly menu system, on-line help, single-pass calculation of estimates for multiple classes of supply, and task force building. Most, although not all, CASCOM desires for improvements are made in the SURE 2.0 system.

In addition, while a superb example of spreadsheet capabilities, the original SURE's limitations make it of limited use for operational planning. It has no provisions for saving parameters for use in creating a logistics estimate for an operational scenario, no provisions for multiple or fractional days, and no ability to correlate estimates with specific phases of an operation. Indeed, a new calculation must be made for every day of an operation where any planning factor changes. In addition to CASCOM requirements and desires, the purpose of developing the SURE 2.0 system under the auspices of the Command and General Staff College (CGSC), and the School of Advanced Military Studies (SAMS) in particular was to make it a better tool for operational planners.

Combining CASCOM requirements to duplicate the functionality of the original SURE system, plus enhancements, with the objective of also making it a tool for operational planners has resulted in a stand-alone system with greatly enhanced capability at the expense of an increase in complexity for even basic operations. Both of these conditions are the result of structuring SURE 2.0 around the concept of orders-based planning.

In SURE 2.0, orders-based planning integrates the concepts of phased operations, parameter sets of operational conditions, and task-organized groupings of units to facilitate the correlation of logistics estimates to operations plans and

2

orders. The framework for the system in an the Order[4], which may be subdivided into as many as nine Phases, each Phase being up to 99.99 days in length. To each order may be assigned any number of Task Organizations, themselves composed of any number of units. For each Phase of the Order, each Task Organization is assigned a Parameter Set, a collection of battlefield and logistical conditions including Area Profile, Combat Intensity, ration cycle, etc.

Although orders-based planning provides enormous flexibility, it does require the user, at a minimum, to create one Unit from an Standard Requirements Code (SRC), one Task Organization from that Unit, one Order with at least one Phase, and one Parameter Set with the appropriate conditions to perform any calculations whatsoever. Considerable effort has been made through the use of on-screen assistance and an additional on-line context sensitive help system to make that process both quick and easy to do. It is, however, different from the original SURE 1.x system, which allows direct manipulation of and calculations based on the SRCs provided. SURE 2.0 preserves the original databases and allows their replacement with updated versions without sacrificing user-developed Units. SURE 2.0 does duplicate 1.x's functionality in terms of the kinds of information calculated and the estimates provided to the user.

One area in which SURE 2.0 was not able to meet design preferences was in minimum system requirements. As a number of older DOS-based machines with only 640K of RAM and small (20 MB) hard drives are still in common use in the Army, it was desired that SURE 2.0 could be run on those platforms. The minimum requirements of dBase IV 2.0 compiled systems, however, is for 640K of base memory and 1MB of extended or expanded memory. When this was identified, CASCOM decided to continue utilizing the dBase compiler, trading the advantages of speed, code maintenance, and ease of use against a more limited set of machines on which the program could be run. In addition to memory requirements, SURE 2.0 also required the 'Files' setting in system Config.sys files to be set to 60 or greater.

3

The SURE 2.0 system started development in November 1993, with projected fielding in late spring or early summer of 1994. The developmental process passed through three general phases: concept development and generation of Alpha code; development of Beta code and Beta testing at Fort Leavenworth's CGSC, and code proofing at Fort Lee. Beta testing at Fort Leavenworth was done by an Advanced Logistics (A459) class in the Command and General Staff College, under the direction of MAJ Richard Simmis of the Deaprtment of Sustainment and Resourcing Operations (DSRO), and by SAMS students during their Battle Command Training Program (BCTP). Upon completion of Beta testing and submission of the code, databases, and this document to the School for Advanced Military Studies the product will be passed on to CASCOM for final proofing and fielding. Future development and code maintenance will be done by CASCOM.

CHAPTER II
PROGRAM STRUCTURE

This chapter is a basic roadmap for future CASCOM programmers doing code maintenance or updates. It reviews the data file listings and general program structure, and is ment to be used in conjunction with the Programmers References in Annex E.

SURE 2.0 allows users to copy and use CASCOM provided databases without modifing the original information. Table 1 below lists the CASCOM databases and their functions (additional data can be found in the Database Structure Summary in Annex E):

TABLE 1
CASCOM AND FM 101-10-1/2 DATABASES[5]

.dbf	Use	Interaction
Ammorate	Ammunition consumption data for all ammunition-using LINs.	Used in calculating ammunition consumption. Related to Csrclinq through lin.
Class_I	Ration types and planning factors.	Used in selection and calculation of ration consumption.
Dodic	Nomenclature, weight, and cube for all ammunition DODICs.	Used in calculating ammunition consumption. Related to Ammorate through dodic.
Echelons	Echelons for water consumption.	SURE attempts to guess Unit echelon, this allows for user correction.
Help	Stores context-sensitive help.	Accessed by F1 key or browsing or printing from Help menu.
Intensty	Intensity and factors for ammunition consumption rates.	Used in calculating ammunition consumption.
Linlist	Equipment from which users can pick to add to their units.	Additions to Csrclinq.dbf based on selections from this list.

.dbf	Use	Interaction
Medical	Class VIII consumption in pounds per man per day. Echelon and intensity peculiar to this database.	Used in calculating Class VIII consumption.
Polrate	Fuel consumption data for all fuel-using LINs.	Used in calculating fuel consumption. Related through pol_type to each other. Polrate is related to Csrclinq through lin.
Polwgt	Pounds per gallon of each fuel type.	
Posture	Ammunition consumption posture.	Used in calculating ammunition consumption.
Profiles	Area Profiles with equipment use hours and kilometers per day.	Used in calculating fuel consumption. Related to Cprofile through profilenam.
Srclist	SRC's from which users can pick to create Units.	Copies to Csrclist, the user's Unit database.
Srclinqt	Equipment LINs for each SRC.	Copies to Csrclinq, the equipment list in the user's Units.
Water	Water consumption per man per day based on climate.	Used in calculating water consumption.

SURE also has an extensive set of databases used for user data, listed below in Table 2:

TABLE 2
USER AND REPORT CREATED DATABASES

.dbf	Use	Interaction
Ammoest	Stores ammunition consumption estimated data.	Created when user selects order for estimation in Orders-based Reports subprogram.
Cl_other	Stores Class I, IV, VIII, and water consumption estimated data.	

.dbf	Use	Interaction
Cprofile	Stores user Parameter Set information, including Area Profile, Intensity, Intensity Factor, Ration Cycle, water and Medical profiles.	Created and edited in Parameter Sets subprogram set. Related to Ordertas through profilenam.
Csrclist	Stores user Units.	Created and edited when user creates Units from SRCs in Units subprogram set.
Csrclinq	Stores user's Unit's equipment LINs and quantities.	Created and edited when user creates Units from SRCs in Units subprogram set.
Orderlst	Stores list of Orders to which Task Organizations with Parameter Sets are assigned.	Created and edited when user creates an Order in the Orders subprogram set. Information drawn from the Taskorg, and Cprofile databases is combined with that from the Order creation and editing screens to create and maintain Ordertas.
Ordertas	Stores Task Organization, strength, and Parameter Set assignment data for each order.	
Polest	Stores fuel consumption estimated data.	Created when user selects order for estimation in Orders-based Reports subprogram.
Taskorg	Stores Unit listings for each Task Organization.	Created and edited when user creates a Task Organization from Units in the Task Organization subprogram set.
Temp	Used by SURE to copy records for temporary processing.	Automatic, no user interaction required.
Ttemp		
Tttemp		

SURE 2.0 contains six major program sets, each defined by a PAD statement in the main sure.prg: Units, Task Organizations, Parameter Sets, Orders, Reports, Utilities, and Help. The following sections review the major functions performed and major sub-programs of each of these program sets. (Additional information on programs and procedures can be found in the Data Dictionary and Program Listings in Annex E.)

```
┌─────────────────────────────────────────────────────────────────────────────┐
│  ┌──────────┐      ┌──────────┐    ┌──────────────────┐    ┌──────────┐      │
│  │ SRCs     │      │ UNITs    │    │ TASK ORGANIZATIONs│    │ ORDERs   │      │
│  │ Srclist  │─────▷│ Csrclist │─▷─▷│      Taskorg      │    │ Ordertas │      │
│  │ Srclinqt │      │ Csrclinq │    │                   │    │ Orderlst │      │
│  │ Linlist  │      └──────────┘    └──────────────────┘    └──────────┘      │
│  └──────────┘                                                                 │
│  ┌──────────┐      ╷··············╷  Task Organizations   Phases             │
│  │ Class_I  │      ┊ User-entered ┊      with Units       Phase Length       │
│  │ Echelons │      ┊ information  ┊                       Strength           │
│  │ Medical  │      ╵··············╵                                          │
│  │ Posture  │      Parameters for each Task                                  │
│  │ Profiles │      Organization, by Phase                                    │
│  │ Intensty │─────▷┌──────────────┐                 ╭───────────╮            │
│  └──────────┘      │PARAMETER SETS│─────────────────▷│           │           │
│  ┌──────────┐      │  Cprofile    │                  │ orderep1.prg           │
│  │ Ammorate │◀─────└──────────────┘─────────────────▷│           │           │
│  │ Dodic    │      Consumption rates,                 ╰───────────╯           │
│  │ Polwgt   │      calculated quantities                   │                 │
│  │ Polrate  │                                              ▽                 │
│  │ Water    │─────────────────────────────────────▷┌──────────────┐          │
│  └──────────┘                                       │   polest     │         │
│                                                     │   ammoest    │         │
│  ╷··············╷  ╷··························╷       │   cl_other   │         │
│  ┊ SURE 2.0    ┊  ┊ SURE 2.0 databases      ┊       └──────────────┘         │
│  ┊ supplied    ┊  ┊ created from supplied   ┊              ▽                 │
│  ┊ databases   ┊  ┊ and user-entered data   ┊       ┌──────────────────┐     │
│  ╵··············╵  ╵··························╵       │Fuel, Ammunition, │     │
│                                                     │and Other Classes │     │
│                                                     │of Supply Estimate│     │
│                                                     │Reports           │     │
│                                                     └──────────────────┘     │
└─────────────────────────────────────────────────────────────────────────────┘
```

Fig. 1 SURE 2.0 Database Interaction

Units

Users are required to create Units from the SRC's provided in the Srclist.dbf, or to assemble them on a custom basis directly from the Linlist.dbf. Although these Units retain their original SRC throughout the system, and can be user-selected for change, addition, or deletion by SRC, SURE uses the unique Unit name assigned by the user. This is required as Units may be based on the same SRC, but, with modifications be quite different. Also, the ability to assign discrete Unit names allows the user to track specific Units throughout an Order estimation report.

Major sub-programs in the Unit program set are in Table 3 below:[6]

TABLE 3
SUB-PROGRAMS IN THE UNIT PROGRAM SETS

Program	Functions
createsr.prg	Creates Units from SRC's
editunit.prg	Edits existing Units
custunit.prg	Creates custom Units from LINs
delunit.prg	Deletes Units

Task Organizations

After at least one Unit has been created, the user may create a Task Organization. Task Organizations also require unique names, although their descriptions may be shared. A Task Organization may have any number of Units assigned to it, and no Unit may be deleted that is assigned to a Task Organization. Task Organizations are one level deep, i.e., only Units can be assigned to Task Organizations - Task Organizations cannot be assigned to each other. Although the desireability of multiple levels of subordination is recognized, it was beyond the scope of this version of SURE do to considerations of duplication and recursiveness.

Major sub-programs in the Task Organization program set are in Table 4 below:

TABLE 4
SUB-PROGRAMS IN THE TASK ORGANIZATION PROGRAM SET

Program	Functions
createta.prg	Creates Task Organizations from Units
createtc.prg	Creates Task Organizations by copying previous ones.
edittask.prg	Edits Task Organizations
deltask.prg	Deletes Task Organizations

The conditions that each Task Organization must operate under during each Phase of an Order are determined by assigning Parameter Sets. Parameter Sets are created by the user by assigning an area profile, intensity and intensity factor, ammunition consumption posture, ration cycle, medical use parameter and water use parameter. These permit complex sets of conditions to be rapidly assigned during the Order creation and editing process. Parameter set names are unique and are not editable after creation.

Major sub-programs in the Parameter Set program set are:

TABLE 5
SUB-PROGRAMS IN THE PARAMETER SET PROGRAM SET

Program	Functions
createpr.prg	Creates Parameter Sets
editprof.prg	Edits Parameter Sets
delprof.prg	Deletes Parameter Sets
crecuspr.prg	Creates Custom Area Profile

Orders

Users establish Orders to define the number of Phases and lengths of each Phase which SURE uses with Parameter Set and strength settings to calculate consumption estimations for each Unit, each Task Organization, and for the Order as a whole. After a Task Organization is assigned to an Order, a Parameter Set is assigned to each Task Organization for each Phase of the Order, and, optionally, the percent strength for each Task Organization for each Phase may also be adjusted.

Strength changes from 100%, if any, are applied to all supply estimates. For example, all Units of a Task Organization which has been assigned an 80% strength for a particular Phase of an Order will have their consumption estimates

reduced by 20% from normal for the same time period. This reduction applies to POL and ammunition estimates as well as to Class I and water estimates for personnel. Its primary application is to approximate those effects anticipated by the Order's manouevre planners.

Major sub-programs in the Orders program set are:

TABLE 6
SUB-PROGRAMS IN THE ORDERS PROGRAM SET

Program	Functions
createor.prg	Creates Orders
editorde.prg	Edits Orders
delorde.prg	Edits Task Organizations

Reports

Estimate calculations based on user Orders, with their associated Task Organizations and Parameter Sets, are made after the user selects an Order in the Orders-based reports subprogram set. After calculations have been made and stored in the Ammoest, Polest, and Cl_other databases the user is offered the choice of printing reports to the screen, to file, or to the printer. Two types of reports are available for fuel and ammunition estimates, LIN by LIN, or summary only. While all reports produce summary estimates for each Unit, for each Task Organization, and for the Order Phase-by-Phase, the LIN-by-LIN option also lists usage for each equipment LIN in each Unit in the POL or ammunition reports. These reports can be quite lengthy, and users are advised to print them to file rather than the screen.

Major sub-programs in the Reports program set are:

11

TABLE 7
SUB-PROGRAMS IN THE REPORTS PROGRAM SET

Program	Functions
orderep1.prg	Calculates logistics estimates and stores them in polest, ammoest, and cl_other databases. This information is accessed through the printpol.prg, printamm.prg, and printoth.prg programs.
sysrep.prg	Allows users to print provided and user-created databases.

Utilities

Several utilities are available through the Utilities subprogram set. SURE 2.0 is compiled under Borland's dBase IV version 2.0 compiler, and the indexes are maintained in .mdx format. Unlike dBase III's .ndx file system, .mdx's which are corrupted or out of date with their associated .dbf files are recognized automatically, and the user is prompted to reindex. Nonetheless, due to the relative newness of the 2.0 compiler the option of manually forcing a reindex of all indexed files in the system has been made.

As not all users will have or be familiar with a general purpose file backup program, simple backup and restore subprograms copy user-created .dbf and .mdx files to a floppy disk. While the system does allow for multiple diskettes, it does not provide for splitting a single file across diskettes. The common use of high density (1.2 and 1.4 MB diskettes) should allow all but the largest of systems to be backed up and restored from this system.

SURE fuel and ammunition reports are similar to those in SURE 1.0, and also require 17-pitch (compressed) print to line up properly when printed. Unfortunately, not all printers are easily adjusted from their control panels to provide this pitch. Three general printer types are supported by SURE to turn this feature on or off in most printers: Epson/IBM compatible dot matrix, HP Laserjet compatible, and HP Deskjet (PCL) compatible. Once the user has made a setting from the 'Compressed Print ON' submenu it is retained and automatically accessed during

the report printing process. In any event, the ability to print to an ASCII file, then import the file to a work processor should overcome the printer problem for virtually all printers.

Major sub-programs in the Utilities Program Set are:

TABLE 8
SUB-PROGRAMS IN THE UTILITIES PROGRAM SET

Program	Functions
reindx.prg	Reindexes (recreates) all .mdx files.
backdbf.prg	Back-up user-created databases and indexes to floppy.
restoref.prg	Restore user databases and indexes from floppy disk.
compress.prg	Turns on compressed (17-pitch) print and sets SURE to automatically print reports in compressed mode where required.
uncompre.prg	Turns off compressed print.

Help

Users are guided through the SURE system's operation in two ways. First, most screens in the system have on-screen instructions from the subprogram's associated inst*.prg procedure file. If this information is not sufficiet, the user can access context-sensitive help by pressing the F1 key. This will access the Help.dbf down to the subprogram set level. The Help.dbf can also be accessed from the Help menu. A larger display of each help subject is used when accessed in this manner, as there is no need for the user to view the current on-screen menu. One or all Help subjects can be printed to the printer. When the Help.dbf is printed this way, the associated menu for a program set is also printed.

Major sub-programs in the Help program set are:

13

TABLE 9
SUB-PROGRAMS IN THE HELP PROGRAM SET

Program	Functions
helpndx.prg	View Help subjects via Help index.
prnhelp.prg	Print one or all Help topics.

Additional, detailed information on fields, programs, and procedures is located in the Tree, Datadict, and Prcsumry sections located in the appendixes.

CHAPTER III
USER HELP

The following section is the SURE 2.0 user help system. It is accessable from SURE 2.0 by pressing the F1 Key, which gets context-sensitive help down to the program-set level on screen, or by accessing the Help menu. The Help menu allows the user to view the Help system by subject, or to print one or all Help subjects. The following sections are, except for formatting, as they would be if printed by a user from the Help menu's 'Print Instructions' sub-program set.

<u>The Help System</u>

```
Browse Help
Print Instructions
```

A. Context-sensitive Help is available in many parts of the SURE System by pressing F1. You may print out the entire Help System from the Main Help Menu. In context sensitive Help, if additional pages are available, a down-arrowwill appear in the lower right corner of the Help frame. If you are in a second or subsequent Help page, an up-arrow will appear. You may transfer between screens using either Up/Down arrows or Page-Up Page-Down Keys. When you are done with Help, press the Enter or Esc Keys to return to the SURE program.

Units Task Organizations Parameter Sets Orders Reports Utilities Help QUIT

B. A Primer on the SURE system.

HOW THE SURE SYSTEM WORKS

Using SURE you create Units from SRC's and their associated equipment LINs, group your Units into Task Organizations, and then assign Task Organizations to an Order. A Parameter Set is assigned to each Task Organization for each

15

Phase of the Order. The SURE system calculates estimated consumption data for each Unit in each Task Organization for every Phase of the Order, and also sums total estimated consumption for each Task Organization and for the Order overall.

** To do any logistical estimations with SURE 2.0 you must have at least one Unit, one Task Organization, one Parameter Set, and one Order! **

UNITS

Units are normally based on SRC's, which are lists of unit sets of equipment in TOE authorized combinations and quantities. Standard SRC's are supplied with the SURE system. Units are usually created by copying SRC listings, then adding or deleting equipment LINs and changing quantities as necessary. Units may also be created from scratch by assigning all equipment LINs and quantities.

Units created from SRC's are given generic names, i.e., 'Infantry Company' if they are to be used generically in multiples, or may be given a specific name, such as '1-17th Infantry' if they are to be used individually and are to have their logistics estimates tracked specifically throughout the Phases of an Order. Once created, Units may be assigned in any quantity to any number of Task Organizations.

TASK ORGANIZATIONS

Units are grouped into Task Organizations which may be assigned to any number of Orders. The SURE system allows you to closely parallel the task organization of units in OPLANs and OPORDERS, although SURE groups of Units are only 'one deep' - you cannot group SURE Task Organizations into other Task Organizations.

A typical use of a SURE Task Organization would be to assign a Headquarters Company and three battalions to it, and call that Task Organization a Brigade. However, if the companies within the battalions need to be cross-attached to

16

create Battalion Task Forces they should be used to create separate Task Organizations (Task Forces) rather than assigning battalions as a Brigade.

MISSION PARAMETER SETS

Parameter Sets are collections of criteria used to calculate logistics requirements. They include combat intensity, ration cycle, and so forth. By creating and saving Parameter Sets with appropriate descriptions, the conditions for each Task Organization in each Phase of an Order can be quickly set or changed.

ORDERS

An Order ties together Task Organizations and Mission Parameter Sets. Each Order consists of up to nine Phases, and each Phase may be up to 99.99 days in length. The conditions established by each Parameter Set are used by SURE in conjunction with the number of personnel and quantity and types of equipment in each Unit to calculate logistics consumption estimates. The use of Units in Task Organizations allows you to produce logistic estimates that parallel the Task Organizations and Phases in an OPORDER or OPLAN.

Also, in order to account for attrition, reconstitution and other changes to Task Organization overall strength, each Task Organization's strength can be adjusted for each Phase of the Order. Logistics estimates areincreased or decreased overall based on the percent strength assigned. If no percentage adjustment is made, SURE uses 100% of Unit personnel and equipment strength for each Phase of the Order.

REPORTS

Two categories of Reports are available, Orders-based Reports which produce logistics estimates, and Systems Reports which allow you to print out listings of data provided with the SURE system as well as listings of the Units, Task Organizations, Parameter Sets, and Orders you have created.

ORDERS-BASED REPORTS

17

Orders-based reports calculate and optionally print logistics estimates. Any changes made to Units, Task Organizations, or Parameter Sets used in an order are automatically reflected in the calculations made.

SYSTEM REPORTS

Systems Reports are useful in planning and adjusting all components of the SURE system. For example, a printed list of the SRC's in the SURE system is useful in planning which ones to select to produce Units. Unit listings facilitate the planning of Task Organizations, while Parameter Set and Task Organization listings help in pre-planning orders.

UTILITIES

Six options are included in SURE's Utilities:

REINDEX FILES

Reindex Files is rarely needed, unless your .MDX files get separated from the rest of your SURE system.

BACKUP FILES

It is highly encouraged to back up your SURE databases and .MDX files to a floppy disk. You should normally do a backup whenever you have done more work than you care to repeat!

RESTORE FILES

This option allows you to copy the files you put on a floppy using the Backup procedure back on to your hard drive.

PRINTER SETUP

Allows you to adjust the printer left margin and page length.

COMPRESSED PRINT ON

Many of SURE's reports require 17-pitch compressed print. If your printer is hard to set manually, SURE will try to do it for you in software.

COMPRESSED PRINT OFF

If you have turned on compressed print from within SURE, this option attempts to reset your printer back to the uncompressed mode.

C1. Creating a Unit

```
Select SRC (Browse)
Select SRC (Find SRC with SRC Number)
Add New Unit from SRC or Copied Unit
Copy Unit from Existing Unit
View Existing Units
Return to Main Menu
```

This program provides several ways to create units for use in your Task-Organizations. You may create a unit by copying its TOE name and equipment listing from an SRC, or by copying its data from a unit you have previously created.

'Select SRC (Browse)'

'Select SRC (Find SRC with SRC Number)'

To select an SRC you may either review the complete list of available SRC's (Browse), or enter all or part of a specific SRC if you have that information (Find SRC with SRC Number).

After selecting an SRC to provide base equipment data for you unit, you may either retain the original, generic TOE description for the unit, or give it a unique name, i.e., '1-57 Infantry Bn'. Units formed using the generic name are useful in planning where you need to use 'type' units, whereas unique names allow you to track logistics estimations for a particular unit throughout the Phases of an Order. Either way, you may only use a unique unit name once in your databases, although you may have any number of that unit assigned to any number of Task Organizations.

19

After selecting an SRC, you must choose 'Create Unit from Selected SRC or Unit' to actually copy the TOE equipment LINs and enter the unit in your unit listing.

'Select SRC (Find SRC with SRC Number)' If you have a list of units with SRC's it is faster to enter them and let SURE locate them. If SURE cannot find an SRC it will automatically start at the beginning of the database and let you Browse to find it.

'Copy Unit from previous Unit' If you are creating several identical or similar units, it may be faster to copy from a previous Unit', then rename and later modify your new unit as required. You must rename the unit if you have copied it from a previous one, as unit names must be unique.

After copying from a previous unit, you must choose Create Unit from Selected SRC or Unit' to actually copy the TOE equipment LINs and enter the unit in your equipment listing.

C2. Editing Unit Data

```
Select Unit - Browse Unit List
Select Unit - By SRC
Change LIN Quantities
Add Equipment LIN
Delete LIN
Change Unit Strength
Change Unit Echelon
Return to Main Menu
```

Units you have added to your unit list may be modified in a variety of ways. Any changes you make to a unit are automatically reflected in all Task Organizations and in all Orders which use that Unit. You must choose a Unit before attempting to make any modifications.

'Select Unit - Browse Unit List' allows you to go through the Unit list inalphabetical order and select a Unit to Edit.

'Select Unit - By SRC' allows you to optionally enter an SRC for SURE to search the Unit list for. This is quite fast if you know even a partial SRC. If you do not enter an SRC, you will Browse the Unit list in SRC order.

'Change LIN Quantities' allows you to increase or decrease the number of items of a particular piece of equipment in a unit.

'Add Equipment LIN' allows you to add items of equipment no already on your Units equipment list. If you accidentally try to add a piece of equipment already on the Unit's list, the program will automatically screen it out.

'Delete LIN' allows you to completely remove a piece of equipment from a Unit's equipment list.

'Change Unit Strength' allows you to set the personnel strength which the program will consider as 100% of the unit's assigned strength.

'Change Unit Echelon' allows you to change the echelon used for water-consumption calculations. This program guessed the echelon when the Unit was first created.

C3. Create Custom Unit from Equipment LINs

```
Enter Unit Data
View/Change LIN Quantities
Add LINs
Delete LIN
Change Unit Echelon
Return to Main Menu
```

'Create a Custom Unit'

The SURE database is regularly updated and contains most unit TOE's. However, if you have a unit which is so unlike any other in the database that it is not even worth copying the equipment data from a similar unit and adjusting it, use this option to create a Unit and, optionally, an SRC for it.

21

'Enter Unit Data' allows you to enter the Unit name, and optionally, an SRC.

'View/Change LIN Quantities' allows you to increase or decrease the number of items of a particular piece of equipment in a unit.

'Add Equipment LIN' allows you to add items of equipment no already on your Units equipment list. If you accidentally try to add a piece of equipment already on the Unit's list, the program will automatically screen it out.

'Delete LIN' allows you to completely remove a piece of equipment from a Unit's equipment list.

'Change Unit Strength' allows you to set the personnel strength which the program will consider as 100% of the unit's assigned strength.

'Change Unit Echelon' allows you to change the echelon used for water consumption calculations. This program guessed the echelon when the Unit was first created.

C4. Deleting a Unit

```
Select Unit - Unit Name Order
Select Unit - SRC Order
Delete Unit
Return to Main Menu
```

'Select Unit' You must select a Unit for Deletion first. Although you may select any Unit, you will not be able to actually delete any Unit that is on any Task Organization listing.

'Delete Unit' After you have selected a Unit for deletion, you can delete it. You will be asked to confirm the deletion. Once deleted, Units cannot be recovered, but must be recreated if required again.

D1. Creating a Task Organization from Units

22

```
Name Task Organization
Assign Units - Unit Name Order
Assign Units - SRC Order
Assign Unit Quantities
Create from Selected Units
Return to Main Menu
```

Task Organizations are groups of Units that will share common Parameter Sets during the Phases of an Order. They could be companies formed into a battalion, or battalions and an HHC formed into a brigade.

'Name Task Organization' Each Task Organization requires a unique name. This name is not changeable, nor may two Task Organizations share the same name. You may however change the Task Organization Description.

'Assign Units - Unit Name Order' After giving the new Task Organization a name, you may select any of the Units in your Unit listing. This option lists them in alphabetical order. After selecting Units, you should assign them quantities of one or more from the 'Assign Unit Quantities' menu.

'Assign Units - SRC Order" If you know the SRC's of the Units you want to add, SURE can find them quickly if you even enter a partial SRC number. After selecting Units, you should assign them quantities of one or more from the 'Assign Unit Quantities' menu.

'Assign Unit Quantities' Units that have a particular identity, such as '1-57 Infantry Bn' should normally be assigned a quantity of one. Generic Units, i.e. 'Infantry Bn', may be assigned in any quantity up to 99 each in any one Task Organization.

'Create from Selected Units' When you are done selecting Units and assigning quantities for this Task Organization, you must Create the TaskOrganization to save the selected data.

D2. Creating a Task Organization by Copying a Previous One

23

```
Select Task Organization to Copy
Assign New Task Organization Name
Save and Copy Another
Return to Main Menu
```

Task Organizations are groups of Units that will share common Parameter Sets during the Phases of an Order. They could be companies formed into a battalion, or battalions and an HHC formed into a brigade.

'Select Task Organization to Copy' If you need a new Task Organization that is similar to one previously created, it is easier to copy the old one and modify it than to create it from scratch. However, each Task Organization requires a unique name. This name is not changeable, nor may two Task Organizations share the same name. You may add or delete units, or change unit quantities to the newly copied Organization from the 'Edit Task Organization' choice on the Main Menu.

'Assign New Task Organization Name' After selecting a previous Task Organization to copy, you must give the new Task Organization a different and unique name.

'Save and Copy Another' You should save your new Task Organization after assigning it a new, unique name. You may continue to copy additional Task Organizations as necessary.

D3. Editing Task Organizations

```
Select Task Organization
Assign Units - Unit Name Order
Assign Units - SRC Order
View Task Organization Units/Change Unit
Quantities - Unit Name Order
View Task Organization Units/Change Unit
Quantities - SRC Order
Un-assign Units - Unit Name Order
Un-assign Units - SRC Order
Return to Main Menu
```

'Select Task Organization' You must have a Task Organization selected for editing before doing any other process.

'Assign Units - Unit Name Order' Any number of Units may be assigned to a Task Organization, in any quantity up to 99 each. Only Units not previously part of the selected Task Organization will be added to it. This option lists Units in alphabetical order.

'Assign Units - SRC Order' Any number of Units may be assigned to a Task Organization, in any quantity up to 99 each. Only units not previously part of the selected Task Organization will be added to it. This option lists Units in SRC Order, and optionally lets you enter an SRC for SURE to find.

'View Task Organization Units/Change Unit Quantities - Unit Name Order' You may view, and optionally edit the quantities of Units assigned to the current Task Organization. This option lists Units in alphabetical order.

'View Task Organization Units/Change Unit Quantities - SRC Order' You may view, and optionally edit the quantities of Units assigned to the current Task Organization. This option lists Units in SRC Order.

'Un-assign Units - Unit Name Order' Units un-assigned from Task Organizations are still part of your Unit listing, and may be added back to the Task Organization later if required. This option lists Units in alphabetical order.

'Un-assign Units - SRC Order' Units un-assigned from Task Organizations are still part of your Unit listing, and may be added back to the Task Organization later if required. This option lists Units in SRC order.

D4. Deleting a Task Organization

```
Select Task Organization
Delete Task Organization
Return to Main Menu
```

25

'Select Task Organization' You must select a Task Organization before-separately attempting to delete it. Task Organizations which have been assigned to an Order cannot be deleted until removed from assignment to all Orders.

'Delete Task Organization' After selecting a Task Organization, you may attempt to delete it. If it is assigned to one or more Orders, you will not be able to delete it. Deleted Task Organizations cannot be un-deleted, and must be recreated if required again.

E1. Creating a Mission Parameter Set

```
Create New Mission Parameter Set
View Mission Parameter Sets
Return to Main Menu
```

'Create Parameter Set' Parameter Sets are groups of settings that instruct the program on what factors to use when doing logistics estimations for Units in a Task Organization during a particular Phase of an Order. You must assign unique names to them, although descriptions may be changed at any time.

When you create a Parameter Set, you will be automatically directed to pick lists for the following items:

Area Profile

You may select from standard Area Profiles supplied with this program, or create non-standard ones from the 'Create Custom Area Profile' choice on the Main Menu.

Ammunition Consumption Posture

Ammunition Consumption Postures are listings of mission conditions, such as '1st Day Defense' that, in conjunction with Combat Intensity are used to estimate the quantities of ammunition consumed.

Combat Intensity

26

Combat Intensity, such as 'Heavy', is used in conjunction with Consumption Posture by modifying that posture through multiplication with the Combat Intensity Factor.

Combat Intensity Factor

A standard Intensity Factor is provided with each Combat Intensity. These are may be modified for a particular Parameter Set, but should be done so with caution as the results will be non-doctrinal.

Class I Ration Cycle

You will select three rations, which will be used by the Parameter Set as the daily ration cycle.

Water Consumption Parameter

These should usually be set according to the echelon of the Task Organization, but may be changed. Larger quantities are calculated for higher echeons.

Class VIII (Medical) Usage Parameter

Echelon and combat intensity are used together to establish a medical usage factor that adjusts Class VIII estimates.

'View Orders' This option allows you to review Orders in the system without making changes.

'Assign Task Organizations' This option allows you to assign Task Organizations to the Order. Only Task Organizations not already assigned will be shown.

'Assign Parameter Sets to Task Organization Phases' Each Task Organization assigned to an Order must have a Parameter Set associated with it for each Phase in the Order. For example, if your Order has three Phases, each Task Organization will have a Parameter Set assigned for each Phase. You may use the

27

same Parameter Set for more than one Phase, and for more than one Task Organization.

E2. Editing a Mission Parameter Set

```
Select Parameter Set
Change Area Profile
Change Ammo Consumption Posture
Change Intensity
Change Intensity Factor
Change Class I Parameters
Change Water Consumption Parameters
Change Class VIII Medical Parameters
Return to Main Menu
```

'Select Parameter Set' You must choose a Parameter Set before making any changes. Changes you make will be reflected in all Orders-based consumption-reports which use that Parameter Set.

'Change Area Profile' You may select from standard Area Profiles supplied with this program, or create non-standard ones from the 'Create Custom Area Profile' choice on the Main Menu.

'Change Ammunition Consumption Posture' Ammunition Consumption Postures are listings of mission conditions, such as '1st Day Defense' that, in conjunction with Combat Intensity are used to estimate the quantities of each ammo DODIC required.

'Change Intensity' Combat Intensity, such as 'Heavy', is used in conjunction with Consumption Posture by modifying that posture through multiplication with the Combat Intensity Factor.

'Change Intensity Factor' A standard Intensity Factor is provided with each Combat Intensity. These are may be modified for a particular Parameter Set, but should be done so with caution as the results will be non-doctrinal.

'Change Class I Parameters' In the Class I Ration Cycle you will select three rations, which will be used by the Parameter Set as the daily ration cycle.

'Change Water Consumption Parameters' These should usually be set according to the echelon of the Task Organization, but may be changed. Larger quantities are calculated for higher echeons.

'Change Class VIII (Medical) Parameter' Echelon and combat intensity are used together to establish a medical usage factor that adjusts Class VIII estimates.

E3. Deleting a Mission Parameter Set

```
Select Parameter Set
Delete Parameter Set
Return to Main Menu
```

'Select Parameter Set' You must select a Parameter Set before attempting to delete it.

'Delete Parameter Set' After you have selected a Parameter Set, you may attempt to delete it. Parameter Sets which are in use by a Task Organization in an Order cannot be deleted.

E4. Create Custom Area Profile

```
Create Custom Area Profile
View Area Profiles
Edit Custom Area Profile
Return to Main Menu

Delete Custom Area Profile
```

'Create Custom Area Profile' Area Profiles establish the distance or timethat fuel burning equipment can expected to be operated for each day. While the standard Profiles supplied with this program are useful in doctrinal calculation, they do not account for exceptional situations, such as a road march or aircraft

flight of known length and time. You may create Custom Area Profiles based on standard Profiles, but you must assign them unique names.

'View Area Profiles' You may view all Profiles, standard and Custom.

'Edit Custom Area Profile' You may only edit Custom Profiles that you have created. If you edit a Profile, future logistics estimates of Orders that use that Profile will automatically reflect the changes.

'Delete Custom Area Profile' You may only delete Custom Profiles that are not in use by any Task Organization in any Order.

F1. Creating an Order

```
Create Order
View Orders
Assign Task Organizations
Assign Parameter Sets to Task Organization Phases
Return to Main Menu
```

'Create Order' Orders combine Task Organizations with Mission Profile Sets. An Order consists of a unique Order number an Order description, and a selection of Task Organizations. An order may have up to nine Phases, each Phase being of up to 99.99 days in length. Each Task Organization is assigned a Mission Profile Set for each Phase of the Order, and may also have its strength adjusted for each Phase.

'View Orders' Prior to creating an Order you may view previous Orders to aid you in assigning a unique Order number.

'Assign Task Organizations' Any number of Task Organizations may be assigned to an Order. Only Task Organizations not previously assigned to the Order will be offered as a selection.

'Assign Parameter Sets to Task Organization Phases' You may assign Parameter Sets to the Task Organizations in an order, one for each Phase, in a number of ways. Combining the following ways can significantly speed up this process.

'Assign one Parameter Set to all Task Organizations and Phases' If all or most of the Task Organizations in this order will be using the same Parameter Set for all or most Phases, use this option to assign a single Parameter Set to all.

'Assign a Parameter Set to a single Task Organization for all Phases'

Use this option if a Task Organization will be using one Parameter Set for all or most Phases.

'Assign a Parameter Set to all Task Organizations for a single Phase.' Use this option if all or most Task Organizations will be using a common Parameter Set in a specific Phase of the Order.

'Assign Parameter Sets to a Task Organization Phase by Phase' This is the most flexible option to assign Parameter Sets to each Phase for a single Task Organization.

'View this Orders Task Organizations and Parameter Sets by Phase' Use this option to view settings without changing them.

F2. Editing an Order

```
Get Order
View/Delete Task Organizations
Assign Task Organizations
Assign Mission Parameter Sets to Task Organization Phase
Assign Strength Percentages to Task Organization Phases
Edit Order Description, Phases, and Phase Lengths
Save and Edit Another
Return to Main Menu
```

```
Assign one Parameter Set to all Task Organizations and Phases
Assign a Parameter Set to one Task Organization for all Phases
Assign a Parameter Set to all Task Organizations for a single Phase
Assign Parameter Sets to a Task Organization Phase by Phase
View this Orders Task Organizations and Parameter Sets by Phase
Return to Previous Menu
```

31

```
Assign one Strength to all Task Organizations and Phases this Order
Assign a Strength to one Task Organization for all Phases this Order
Assign a Strength to all Task Organizations for a single Phase this Order
Assign Strengths to a Task Organization Phase by Phase for this Order
View this Orders Task Organizations and Strengths by Phase
Return to Previous Menu
```

'Get Order' You must choose an order before making any changes.

'View/Delete Task Organizations' Use this option to review the names of all Task Organizations already assigned to this order, and optionally, delete them. Task Organizations deleted from an Order are still in the SURE system, and may be added back, but you will have to assign Parameter Sets again.

'Assign Task Organizations' You may add additional Task Organizations to your Order. Only Task Organizations not already assigned to the Order will be shown.

'Assign Mission Parameter Sets to Task Organization Phases' Each Task Organization must have a Parameter Set assigned for each Phase of the Order.

'Assign Strength Percentages to Task Organization Phases' Each Units equipment and assigned personnel are used as 100% strength unless you change this figure for the Task Organization for a specific phase of an Order. For example, reducing a Task Organizations strength by 90% for a particular Phase would also reduce all logistics estimates by 10% for that Phase for all Units in that Task Organization.

'Edit Order Description, Phases, and Phase Lengths' Although you may change the number of Phases in an Order, if you increase them, you will also have to assign Parameter Sets for each Task Organization for each Phase added.

'Save and Edit Another' Changes are made as you enter them, but this option resets this subsystem to assist you in making changes to another order.

F3. Deleting an Order

```
Get Order
Delete Order
Return to Main Menu
```

'Get Order' You must first select an Order before having the option to Delete it.

'Delete Order' If you delete an Order, you will not be able to un-delete it and get it back!

G1. Orders-based Reports

```
Select Order/Calcuate Consumption
POL Estimate
Ammo Estimate
Other Supply Estimates
Return to Main Menu
```

```
Print Estimate to Printer
Print Estimate to Screen
Print Estimate to ASCII File
Change Printer Defaults
Return to Previous Menu
```

```
LIN by LIN Report, plus all Totals
Summary Report - Unit, Task Org, and Order Totals
Summary Report - Task Org and Order Totals Only
Summary Report - Order Totals Only
```

Orders-based Reports produce logistic's estimates from your Units, Task-Organizations, Parameter Sets, and Orders. You will be offered a choice to print them to the printer, to your computer screen, or to a an ASCII (plain text) file which can be imported, edited, and printed with any workprocessor.

'Select Order/Calculate Consumption' You must select an Order, after which basic calculations for consumption data will be automatically performed.

'POL Estimate' Bulk POL consumption estimates are automatically made by equipment LIN, by Unit, by Task Organization, and for the Order as a whole, divided by Phase. You will have the option to print the estimate to the printer, a

33

file, or the screen and to print out estimates LIN by LIN, or summations by Unit, Task Organization, and Order only. As this report can be quite long, printing to a file is recommended. The ASCII text file created can be imported into any word processor for viewing or printing.

'Ammo Estimate' Ammunition consumption estimates are automatically made by DODIC, equipment LIN, Unit, Task Organization, and for the order as a whole, divided by Phase. You will have the option to print the estimate to the printer, a file, or the screen and to print out estimates LIN by LIN, or as DOCIC summations by Unit, Task Organization, and for the Order as a whole. As this report can be quite lone, printing to a file is recommended. The ASCII text file created can be imported into any word processor for viewing or printing.

'Other Supply Estimates' Class I, Water, Class IV and Class VIII consumption estimates are made for each Unit, and summed by Task Organization and for the Order as a whole. You will have the option to print the estimate to the printer, a file, or the screen. As this report can be quite long, printing to a file is recommended. The ASCII test file created can be imported into any work processor for viewing or printing.

POL Reports

SURE Bulk POL Reports list estimated bulk consumption by Unit, Task Organization, Phase of the Order, and Order Total.

If you choose the 'LIN by LIN' option the report will list fuel consumption data and fuel tank capacity for each fuel-using LIN in each unit. Fuel tank capacity is for internal storage only - a bulk POL tanker, for example, will NOT have its bulk load included in this figure. After the LIN reports, you will get total fuel projections by Unit, Task Organization by Phase, Phase of the Order, and Order Total reports.

If you choose one of the 'Summary Report' options you will not get LIN's and their individual consumptions. You will still get Unit, Task Organizationby

Phase, Phase of the Order, and Order Total reports according to the option you choose. As these reports can be quite lengthly, the 'Summary Report - Task Org and Order Totals Only' choice may be your best compromise between detailed information and length.

All estimates are adjusted for any changes to strength percentage you may have made for a particular Task Organization in a particular Phase of the Order.

Ammunition Reports

SURE Ammunition Reports list estimated ammunition consumption by Unit, Task Organization, Phase of the Order, and Order Total.

If you choose the 'LIN by LIN' option the report will list DODIC consumption data and basic loads for each DODIC-using LIN in each Unit. Weights and cubes shown are for ammunition consumed, not for basic loads. After the LIN reports by DODIC you will get DODIC wrap-ups by Unit, Task Organization by Phase, Phase of the Order, and Order Total.

If you choose one of the 'Summary Report' options you will not get LIN's and their individual consumptions. You will still get Unit, Task Organizationby Phase, Phase of the Order, and Order Total reports according to the option you choose. As these reports can be quite lengthly, the 'Summary Report - Task Org and Order Totals Only' choice may be your best compromise between detailed information and length.

All estimates are adjusted for any changes to strength percentage you may have made for a particular Task Organization in a particular Phase of the Order.

'Other Supply Estimates'

In addition to POL and ammunition reports, SURE can also produce estimates for Rations, Water, Barrier/Construction material, and Medical Supplies (Class VIII). These are based on the number of personnel authorized to each Unit,

adjusted by any strength percentage change you may have made to the Unit's Task Organization for any particular Phase of the Order.

Rations are by ration type, in number of meals, water is in gallons, barrier/construction material is in pounds, and medical supplies are in pounds. Estimates are given for each Unit, for Task Organizations as a whole by Phase, for each Phase of the Order, and for the Order Total.

G2. Systems Reports

```
SRCs (TOE Listings)
SRC with LINs
Units
Units with LINs
Parameter Sets
Task Organizations
Task Organizations with Units
Orders
Orders with Task Organizations and Profiles
Orders with Task Orgs, Units, and Profiles
Change Printer Defaults
Return to Main Menu
```

'SRCs (TOE Listings)' A complete list of all SRC's supplied with the SURE system. This listing is useful in planning the Units you need to create, particularly as many SRC's are similar.

'SRC with LINs' You may select one SRC and print it with all of the TOE equipment items and quantities.

'Units' You may print a complete list of all Units you have created within the SURE system. This list is useful is planning Task Organizations.

'Units with LINs' You may print a listing of your Units with their assigned equipment LINs and quantities.

'Parameter Sets' You may print a list of all of the Parameter Sets you have created with their assigned settings.

'Task Organizations' You may print a list of all Task Organizations you have created within the SURE system.

'Task Organizations with Units' You may print a list of all Task Organizations with their assigned Units.

'Orders' You may print a list of all Orders you have created within the SURE system.

'Orders with Task Organizations and Profiles' You may print a list of all Orders with their Task Organizations and Profiles.

'Orders with Task Orgs, Units, and Profiles' You may print a list of all Orders with their Task Organizations, the Units assigned to those Task Organizations, and the Profiles assigned to each Task Organization for each Phase of the Order.

'Change Printer Defaults' You may change printer page width and page length.

```
Reindex Files
Backup Files
Restore Files
Printer Setup
Compressed Print ON
Compressed Print OFF
```

'Reindex Files' Reindexing recreates all indexes used by SURE. It is not normally needed, as SURE's compiled dBase 2.0 system should automatically prompt you if it needs to repair an index. However, if your .MDX index files should become deleted and you do not have a current backup, this will recreate the .MDX index files. Do NOT use this option without a reason!

'Backup Files' You may backup your SURE files to a floppy disk. You must have a blank, formatted floppy to start. If you use a floppy containing a previous SURE backup your old files will be overwritten.

'Restore Files' If you have a floppy disk with files from a backed-up SURE system you may Restore them to this computer. If you do so, your current files will be overwritten, and all data entered in this system will be lost and re-placed from the files on the floppy disk. It is HIGHLY recommended that you do a backup of this system first before doing a Restore.

'Printer Setup' Adjust left printer margin and page length. The type of printer for compressed print, if any, is also displayed.

'Compressed Print ON' Many SURE reports must be printed in 17-pitch, compressed print. If yourprinter does not have a panel switch SURE will try to set the option for you. Once this option is set, SURE will remember what printer type was choosen and automatically select compressed print for reports where it is required.

'Compressed Print OFF' If you have set compressed print on from the SURE system, this option will try to restore your printer to normal print.

CHAPTER IV
LIMITATIONS AND RECOMMENDATIONS

Although a radical improvement over the previous SURE 1.0 system, SURE 2.0 suffers from technical, developmental, and doctrinal limitations that should be overcome in future versions.

The dBase IV 2.0 Compiler For Dos

SURE 2.0's technical limitations are the result of using Borland's dBase IV 2.0 environment for development and fielding. While an improvement in both features and speed over previous versions of dBase, this DOS based version can neither run on 'minimal' machines --those with 640K RAM-- nor provides the advantages of a Graphical User Interface (GUI). As a result, the advantages of event-driven programming --mice, slider bars, transparent printer and display handling, etc.-- in a Windows or OS/2 environment are unavailable to the user, and Computer Aided Software Engineering (CASE) tools such as those in Computer Associates dBfast are unavailable to the programmer.

SURE 2.0 does posses the advantage of being able to run on merely obsolescent machines with 2MB of RAM, and without a mouse, making it suitable for use on older machines, particularly laptops, many of which are still in the system. However, as SURE 1.0 would run on virtually any DOS machine with a copy of an appropriate version of Lotus Developments 1-2-3™, future versions of SURE could and should be ported to a GUI environment. dBfast™ for Windows™ from Computer Associates is currently available for Microsoft Windows™, and an OS/2™ version is expected.

Procedural Limitations

Limitations of Task Organizing only one-level deep.

As noted in this paper's Introduction, SURE 2.0 has significant value in facilitating the logisticians efforts to perform parallel planning along with his maneuver counterpart as a result of its orders-based planning paradigm. Although utilization of the '2 levels down' planning concept makes SURE 2.0 valid for many operational scenarios, considerable additional utility would be gained by adding the capability to assign Task Organizations to other Task Organizations. This was one of the desires, although not a requirement of CASCOM's SURE 2.0 proposal, and is highly desirable for the next reiteration of the program.

Doctrinal Limitations

Lack of a single CSS approach to evaluation of the battlefield.

Programming the SURE system revealed clearly what was hidden in the complexity of FM 101-10-1/2: CSS planners do not have a uniform way of analyzing the battlefield, and insufficient guidance to confidently use the ways that are established. Separate parameters must be set for ammunition, fuel, water, and medical calculations only make sense if they relate to the nature of the material being issued, not the nature of the battlefield.

Users must now select from a bewildering, uncoordinated set of choices that need resolution in the near term. In particular, the CSS community needs to develop a uniform set of standards for evaluating battlefield conditions, promoting both a common set of terms and a communality of understanding of what the terms mean. What are the precise indicators for 'Heavy' combat? Are they different in different theaters, or for different types of units?

Limited use of some outputs.

Although in accordance with doctrine, SURE 2.0's outputs for Class IV and Class VIII material are of very limited use. Pounds of barrier, construction, and especially medical material may be useful for TRANSCOM or for planners at theater level, but such figures are incongruous in a program that estimates the number of rounds of 9mm ammunition and their weight to two decimal places! A knowledge of geographical area, weather, and mission conditions could and should be used to estimate quantities of sandbags, 4X4's, sheets of 3/4" marine plywood, pickets, Mod Med kits, units of blood, etc.

Conclusion

The SURE 2.x will likely continue for several years as a compiled dBase system running in the DOS environment. Its weaknesses reflect a lack of consistency within the CSS community, and should be addressed in doctrine and then in the program's capabilities. SURE 2.0 is not an extremely simple program to use, but it makes available of wealth of reports in a level of detail which would be highly impractible to do manually, and it provides extensive Help systems to coach the new user through the process.

ENDNOTES

[1] *FM 101-10-1/2*, p. 2-0.

[2] As of this writing, the original SURE worksheets continue to be updated, the latest version being 1.1. As they are essentially similar, all worksheet versions are refered to as '1.x.'

[3] U.S. Army, *'Supply Usage Requirements Estimator (SURE) Version 2.0 (Compiled Version)*, (Fort Lee: CASCOM, not dated), p. 1.

[4] The SURE 2.0 convention of capitilization of major SURE elements is followed throughout this document. These include all user-created elements: Unit, Task Organization, Parameter Set, and Order.

[5] Databases are shown with initial capitals, i.e. 'Csrclinq,' while database fields are lowercase.

[6] More detail on SURE 2.0 program, database, and procedure relationships can be found in the appendices under the Tree Diagram.

[7] Parameter Sets were originally refered to as Mission Profiles, and many of the procedure names reflect this earlier coding. The name was changed to Parameter Sets to avoid user confusion with Area Profiles, which are only one element in a Parameter Set.

ANNEX A

CASCOM SURE 2.0 REQUIREMENTS

Supply Usage Requirements Estimator (SURE)
Version 2.0 (Compiled Version)

General Specifications

1. Background. The Supply Usage Requirements Estimator (SURE) consists of a series of Lotus 1-2-3 macro-driven programs designed to assist operational planners is estimating supply requirements for Army units. The current version of the program (Version 1.01, Sep 93) allows planners to estimate detailed equipment-based consumption for supply classes III and V, and population-based consumption for classes I, II, IV, VI, and VIII, and water. The spreadsheets included with SURE contain a number of databases upon which calculations are based. The major databases include:

  a. SRC, Designation, Strength.

  b. Equipment Density by SRC.

  c. Ammunition Rates by LIN and DODIC for a number of postures.

  d. POL Rates by LIN.

  e. Standard Equipment Usage Profiles for a number of geographic locations.

  f. Factors for population based consumption.

  g. Various factors pertaining to combat loads, combat intensities, climate, unit echelon, theater.

These databases, combined with the macros contained within the spreadsheets, provide a means by which the user can automatically calculate the desired supply estimates within a matter minutes.

2. Limitations. Although the current version of SURE puts an impressive capability within the hands of the user, it does have a significant limitation in that it requires Lotus 1-2-3 Version 2.x in order to run properly. This forces the SURE spreadsheets to be limited by the memory constraints imposed by Lotus 1-2-3. In addition, this requires each potential SURE user to own or purchase a copy of Lotus 1-2-3. The matter is complicated by the fact that there are several versions of Lotus 1-2-3 available (2.x, 3.x, Windows) as well as versions of competing software (e.g., Excel, Quattro Pro) which may or may not process Lotus macros. Many locations have standardized on another product as their preferred spreadsheet. As a result, many potential users are unable to run SURE without first incurring an additional expense.

3. Compiled SURE. With the above limitations putting a ceiling on the expansion of SURE, CASCOM (FD&E) has decided that the next major version of SURE should be totally reprogrammed as a database, as opposed to spreadsheet, application. Further, this database application will be programmed in a language that can be compiled to a stand-alone application, therefore not requiring users to possess any other application software in order to run. This new version of SURE will possess, at bare minimum, all the functionality of the Lotus-based software. In all likelihood, the functionality will be greatly enhanced to include features described below.

4. Features. The following is a list of desired hardware and software features that apply to the new SURE program. These are not to be considered hard and fast requirements, but rather "goals" which may require refinement as the programming effort takes place.

    a. Hardware Environment. The compiled SURE program should be capable of being run on a minimum configuration of an IBM-compatible 80286-based PC with 640K of RAM, a 20MB Hard Drive, and MS-DOS 3.x. A color monitor cannot be assumed, as monochrome monitors and liquid crystal displays (LCD) may still be in use. (Although the initial release of the compiled SURE program will be a DOS-based program, future development may include a Windows-based version as well. This, of course, would set a different minimum for hardware requirements.)

    b. Compiler. The SURE program must be compiled to a stand-alone application. The dependency upon additional run-time software is not acceptable unless the compiler licensee specifically grants the developer the rights to unlimited distribution of the run-time software. Preferred development environment is Borland's dBASE Compiler Version 2.0. This product creates an EXE file which will run in a stand-alone mode. The Borland environment has the advantage of using the standard dBASE IV Version 2.0 interpretive package as a development and testing tool.

    c. Documentation. All source code will be internally documented through the liberal use of comments. Meaningful variable names will also be used to enhance the readability and understandability of the source code. All code developed will remain the property of the U.S. Army and, in particular, CASCOM. Appropriate user documentation will be developed as well.

    d. File Structures. All data files will use the dBASE-compatible .DBF file structure. All rates, profiles, factors, unit data, etc., will be contained in data files, as opposed to being embedded within the source code. This structure is required so that updates to the data can be easily made without recompiling the

program. Thus, users may receive updated data disks without having to update the program.

e. User Interface. The user interface should be constructed so that the program is virtually self-explanatory. Context-sensitive help screens and functions are greatly desired. The program should employ, if at all possible, the techniques of scrolling pick lists (e.g., for selecting an SRC to use), radio buttons (e.g., for selecting only one of many echelons, intensities, profiles, postures, climates, etc.), and check boxes (e.g., in instances where more than one of many choices can be made). Programming for optional mouse usage is also desirable.

f. Colors. Liberal use of colors is desirable, but the colors chosen should be of sufficient contrast so that the program can be used on monochrome and LCD screens.

g. Output. Users should be allowed to direct output to either the screen or the printer. Existing SURE report formats are to be considered adequate for the first cut of the compiled version.

h. Integration. The current version of SURE requires the user to make separate passes through the program for Class III, Class V and the population based factors. The compiled SURE should integrate this process so that a user need make only one pass through the program to calculate estimates of any or all of the available classes of supply. Also, the population based portion of compiled SURE should draw default population figures from the SRC/task force used for the other classes of supply.

i. SRC Selection. When a user is expected to enter an SRC, they should have the choice of either entering the SRC directly or scrolling through the list of available SRCs in order to select the one desired.

j. Task Force Building. The user should be able to incorporate an existing task force within a new task force. For example, if a user is constructing a division-sized task force, he or she should be able to draw upon any brigade-sized task forces they developed earlier. Adding this feature may require a validation mechanism to prevent a user from creating a circular relationship.

k. Save Parameters. Given that a user will have to provide a set of parameters in order to initiate a SURE run (e.g., profile, posture, intensity, feeding policy, climate, echelon, theater, etc.), the program should allow the user to name and save these parameters for future use, thus avoiding having to reenter them. Users should be able to retrieve, modify, copy and delete parameters as needed.

l. Scenario Building. A desired (but not absolutely required) feature of the first version of compiled SURE is the ability to create a "scenario" over time by varying the sets of parameters applied to a unit, set of units, and/or task force.

m. Other Classes of Supply. It is highly likely that sufficient data and methodology will be available in order to incorporate classes VII and IX into the first compiled version of SURE. The programming effort should be structured to allow for this probability.

n. Profile Building. Users should be able to build their own equipment usage profiles that can be used in lieu of the ones provided with the SURE program.

o. Menu Structure. Menus should be logically structured so that the return path from a nested menu selection is the reverse of the path taken to make the selection in the first place.

5. In-Process Reviews. Once initiated, the work in progress will be subject to semi-monthly in-process reviews, with the first review tentatively scheduled for January 15, 1994. These reviews will consist of assessing and testing the program developed to that point and making recommendations for changes and/or further development. It is CASCOM's goal to have a compiled version of SURE ready for distribution by June 30, 1994. Therefore, a final draft working copy of the program should be available no later than May 15, 1994. Should the program be developed to include other classes of supply, the necessary logic, algorithms and data bases will be made available no later than February 28, 1994.

6. Summary. The intent of this document is to lay down the framework for the compiled SURE program. There is enough detail contained herein to provide the basis for discussion regarding the SURE programming effort. More detailed specifications will be prepared as required.

ANNEX B

MEMORANDUM OF UNDERSTANDING BETWEEN CASCOM
AND
THE SCHOOL FOR ADVANCED MILITARY STUDIES

MEMORANDUM OF UNDERSTANDING
BETWEEN
CASCOM AND SAMS

FOR: Commander, U.S. Army Combined Arms Support Command (CASCOM), ATTN: ATCL-FSP, Fort Lee, Virginia 23801-6000

Director, The School for Advanced Military Studies (SAMS), Command and General Staff College, Fort Leavenworth, Kansas 66027
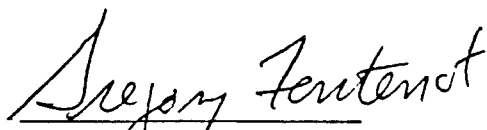
SUBJECT: Agreement between Commander, CASCOM and Director, SAMS for the production of SURE 2.0.

1. PURPOSE: This memorandum stipulates the terms and conditions for production of *Supply Usage Requirements Estimator* (SURE) 2.0 as a database-based, stand-alone replacement for the current 1-2-3-based, macro driven version.

2. AGENCIES: The U.S. Army Combined Arms Support Command (CASCOM) is the logistics planning factors manager for the Army. Within CASCOM, the planning factors mission is assigned to the Planning Factors Management Branch, Force Development and Evaluation Directorate. The Logistics Data Base (LDB), developed by the CASCOM to store, process, and disseminate data for all classes of supply, is physically located at the Directorate of Information Management, Information Systems Command, Fort Leavenworth, Kansas. The School for Advanced Military Studies (SAMS), is an element of the Command and General Staff College, Fort Leavenworth, KS.

3. RESPONSIBILITIES: CASCOM will provide SURE 1.0, basic databases in .dbf format, a copy of dBase 2.0 for a programming environment, and will specify minimum requirements for SURE 2.0. SAMS will provide a student/programmer who will be awarded credit for one monograph for this effort, and will insure structured direction of the project through use of one or more monograph directors.

4. MILESTONES:

ITEM	PROJECTED COMPLETION	STATUS
CASCOM Visit	5-7 December 1993	Completed 5-7 Dec 93
Prospectus Turn-in	24 January 1994	
Complete CASCOM MOU	24 January 1994	
First IPR (Program Structure)	24 January 1994	
Second IPR (ALPHA Code)	23 February 1994	
Third IPR (BETA Code) [CASCOM Visit]	22 March 1994	
Monograph Final - Release Candidate Code	20 April 1994	
Monograph Final to DGDP	4 May 1994	
Program Release by CASCOM	TBD	

5. EFFECTIVE DATE. This agreement is effective from 24 January 1994 until release of program in May, 1994.

FOR THE COMMANDER:

GREGORY FONTENOT
COL, AR
Director, SAMS

MICHAEL E. VELTEN
COL, TC
Director, Force Development
and Evaluation

16 Feb 9

ANNEX C

DEVELOPMENTAL NOTES

CASCOM Interim Comments

In addition to weekly telephone contact with Mr. Bob Miller, two sets of interim comments were made by CASCOM during SURE 2.0 development. The first, dated 17 February 1994, was received from Mr. Miller early enough in the development process to be acted upon.

The second, dated 18 March 1994, was received in a meeting with Mr. Alex Blair. It included some items not part of the original specifications or MOU. This document is divided into three areas, 'Minimum Essential Requirements,' 'Features That Are Highly Desirable for an Effective SURE V 2.0,' and 'Features That Are Desirable, but Not Necessary for an Effective SURE V 2.0.'

Most of the new 'Minimum Essential Requirements' of the 18 March document were met, with the primary exceptions of 'backing-out' from all SURE processes using the ESC key. Although this is technically possible, complete re-coding of virtually all menus would be required, an unfeasible request ninety days and 10,000 lines into the program. A compromise was reached concerning the use of SRC's vs. Units, where both capabilities would be included when selecting or editing.

Under the 'Highly Desirable' listing, use of the Enter key to move forward was included for all possible browse screens. Task Force (Task Organization) within a Task Force capability was not, for reasons covered in the main body of the monograph. Menu's were not restructured, although additional on-screen Help,

plus a context-sensitive Help system now facilitate user use of an extremely capable, though complex system.

'Desirable by Not Necessary' features were retained. 'Scenario Building,' as CASCOM terms it, is the orders-based process that is at the heart of the SURE 2.0 system. Context-sensitive Help was added and has been retained.

SURE Version 2.0 Comments - based on build of 15 Feb 94

16-17 Feb 94

John,

You've done an incredible amount of work in a very short period of time! I appreciate the frequent updates of the code, and I regret that I haven't been able to provide detailed feedback for you until now. I will try to provide you as much detail as I can as I go through the program. (I will be creating my comments in one window while I'm running the program in another.)

Before getting into some nitty gritty specifics, I'd like to suggest some generalized course corrections. Once these are in place, I think it will be easier to address some of the specifics.

The thing we need to keep in mind throughout the development of the SURE program is who the user is and what he or she expects. I think we must assume that the typical SURE user will be computer literate to the extent that they are able to run applications on the computer. For example, with the current version of SURE, the user must be literate enough to know how to load Lotus 1-2-3 and how to retrieve the AASTART spreadsheet. They are not, however, expected to be familiar with the underlying Lotus 1-2-3 program. Similarly, with Version 2 of SURE, we can expect the user to be literate enough to know how to start the SURE application. However, we cannot assume that they have any familiarity at all with the underlying dBASE program.

Therefore, it is of utmost importance that we keep the application as simple to use as possible and the keystrokes in line with what will probably be familiar based on other applications. For example, despite the idiosyncrasies of dBASE, we need to try to establish that ENTER is used to make a positive selection and ESCape is used to abort and go back one step. Similarly, if help is employed, it should be tied to the F1 key. HOME should be used to take a user to the top of a list (in lieu of dBASE's Ctrl-PgUp) and END should be used to get to the bottom. To summarize, let's use the following as "standard" keystrokes for the following activities:

ENTER	Used to select an option from a list
ESC	Used to cancel out of current operation (abort)
	and/or to go back one level, menu, etc.
HOME	Used to go to the top of a list
END	Used to go to a bottom of a list

Also, it is extremely important to make the program as "idiot-proof" as possible. (I realize that that is impossible because idiots are so ingenious!) This means that as you develop your code, try to see how it responds when the user hits the wrong key. As we get further along in program development, I will be trying to help you crash-proof the SURE program as much as possible.

One other thought of a general nature. When you want to take a break from programming, take time to run through the existing SURE program so that you know how it works, what the user is expected to do, and what the reports look like. Even though SURE V 2 is a total rewrite using different technology, from the user's point of view they don't want to have to completely relearn the whole program. So to the extent that the Lotus version of SURE is doing something right, we need to try and parallel that feature as much as possible in the compiled version. This is particularly true with respect to reports.

With all this in mind, let's get to the program!...

I like the way you have the main screen laid out. I also like the recent addition of the explanation for the menu options. We spoke earlier about colors, and you said you preferred to work in black & white for fear of losing contrast on a laptop's liquid crystal screen. It may seem insignificant, but I think aesthetics are very important (especially when "showing off" the program to the chain-of-command!), so let's try some color and see what happens. Here's my suggestion:

```
IF ISCOLOR()
        SET COLOR TO W+/B,B/W+
ENDIF
```

If this combination poses a problem for laptops, we can fine-tune later. I've got a laptop here at work that I can use to test things on.

Just out of curiosity, what is the purpose of the memory statistic on the next to the last line?

The way you have the menus set up is just fine. As we get further down the road, I may ask to have you rearrange the order of the major menu options to put the most frequent activities up front. I will be gathering input from people most familiar with the current SURE program and providing feedback at a later date.

Now let's deal with the "Unit" menu itself. One option we should add is the ability to Create a Unit from scratch. The user should be able to enter their own SRC, title and strength, and then select LINs from the available list.

Now I'm going to work with the "Create Unit from SRC Option". First of all (and this really applies to all submenus, not just this one), I believe we need to incorporate a "Return to Previous Menu" option in each of the submenus in order to allow the user to back up. You may wish to also tie in the ESC key to perform the same function. The first two options on this menu can actually be combined into one. When a user decides to Create a custom unit from an SRC they can be prompted to enter all or part of an SRC number. If they leave the field blank they will simply be sent to the top of the file. Your second option essentially works this way already. We probably should use the word "Custom" or some synonym thereof to distinguish a user's SRCs from those provided with the program. Your "DONE!" option works fine if the user wishes to move forward. But if they've selected a unit and then decide that they want to quit the operation, the DONE will save the selected unit whether they want to or not. My suggestion is to remove the DONE options from the menu and to incorporate a

54

"Do you wish to save this unit?" question into the actual unit creation routine. This question could be followed by a "Do You Wish to Create Another Custom Unit?" question. The idea here is to limit the menus to a series of options which are INDEPENDENT of each other. If one menu option is dependent upon a previous one (as we'll see in some of the other menus), then I would suggest that these not be menu options at all. To recap, for this particular menu, I would suggest the following as the available options:

> Create Custom Unit from SRC
> Create Custom Unit from Existing Custom Unit
> View Existing Custom Units
> Return to Previous Menu

Further, while sitting on the menu, I would recommend making the ESC key active and, if pressed, perform the same action as the "Return to Previous Menu" option. These suggestions can be considered applicable to most of the other menus as well.

Now I'm pressing the "Create Unit from SRC (Find SRC)" option. I like the way the look-up works. You may wish to expand the user prompt to explain that they may enter all or part of an existing SRC (or none at all to go to the top of the list). I notice that when I repeat this option, the previous field value is returned. Suggest you blank out the user input field each time. Also suggest that if a user enters an exact match there is not need to throw them into the browse screen. It should skip that step.

Now we get to the browse screens. As we discussed on the phone, and as I generalized above, it is imperative that we use the ENTER key to move forward and the ESC key to back up. I wrote some sample code which demonstrates the technique that we discussed last week.

```
DEFINE WINDOW W_BROWSE FROM 5, 10 TO 20, 70
CLEAR TYPEAHEAD
ON KEY LABEL CTRL-M KEYBOARD CHR(23)
ON KEY LABEL HOME KEYBOARD "{CTRL-PGUP}"
ON KEY LABEL END KEYBOARD "{CTRL-PGDN}"
USE C:\SURE20\SRCLIST
ACTIVATE WINDOW W_BROWSE
BROWSE NOAPPEND NOMENU NOEDIT NODELETE COMPRESS FREEZE SRC
ON KEY LABEL CTRL-M
ON KEY LABEL HOME
ON KEY LABEL END
DEACTIVATE WINDOW W_BROWSE
IF LASTKEY() = 27
        ? "YOU PRESSED ESC"
ELSE
        DISP
ENDIF
```

55

This sample remaps the ENTER key to Ctrl-End, the HOME key to Ctrl-PageUp and the END key to Ctrl-PageDn. It also sets up the browse window so that the cursor is "frozen" in the SRC field. You'll also note that I wrote a trap for the ESC key so that the "quit and go back" logic can be incorporated.

In a nutshell, we MUST incorporate this logic (or something similar thereto) into ALL browse screens.

On this particular browse screen, we need a prompt to instruct the user what to do.

The next screen (change the unit name and strength) has me a little confused. It appears that you are using the unit name field as a key field for identifying a unit. (I'm basing this on the paragraph of instruction that you provide with the prompt.) The key field for any unit (whether standard or custom) must be the SRC field. The user should be prompted to enter their own (unique) SRC number to distinguish it from the original. (The current SURE program allows them to change the last four positions.) The number should be checked for uniqueness (against all existing and custom SRCs) before being accepted. The Ctrl-W prompt is not really necessary on this screen. However, after the user enters their SRC number and changes the title and strength (as appropriate), they should probably be given the option to change Line Item information within the unit. (I realize that you can do it by going up two menus and then back down again, but this is a little awkward.) Once the information is entered, the user should be asked if they want to save the custom SRC. If they answer yes, then the records are saved. Whether they answer yes or no to the first questions, they should then be asked whether they want to create another custom SRC.

In your current menu, when you press DONE (either one), the "Please wait..." message appears on top of the preceding message.

OK; now I'm going back up one level and down to the "Edit Unit" option of the "Unit" Menu. When I choose the "Select Unit" option, the browse screen only has the unit titles. This should include the SRC and the strength. Also, as already discussed, selection should be made with the ENTER key. ESC should back out without selecting anything. Once you've selected a unit, you display the information at the bottom of the screen. Suggest the include the unit strength so that the user will know whether he wants to change it or not.

Actually, the header information from the SRC (title, strength, and echelon) could probably be addressed in one @ SAY... GET screen with a READ statement. (Speaking of echelon, I notice that you are providing for it as you "edit" a unit, but you haven't provided for it as you create a unit.) Of course, I'm not certain at this point what your intent is for establishing an echelon, because the original SRC list does not allow for them.

On the "Change LIN Quantities" screen, it might be better to show only one nomenclature and then have a check mark in a column indicating whether it is an ammo user or a POL user. As we fine-tune the program, we might want to come up with a master LIN file which has LIN, NOMEN, POL (T/F), and AMMO (T/F) as field names. Then, as we add other LIN-dependent classes of supply, we can add fields for Class IX (T/F), etc. If you like, I can create a file like that within a very short period of time.

56

Your prompt at the bottom of the screen should be changed to reflect that the user is expected to actually change quantities on the screen rather than "choose". Also, since this type of screen is different from previous browse screens (in that the user is allowed to change information), we should establish a "standard" for its use. Obviously, ENTER may used to enter a quantity, so it could pose a problem if used to exit the browse. I would prefer not to rely on ESC to get out of browse because of what I said earlier, and I would like to reserve HOME and END to go to the top and bottom of the file. (As you can see, I'm brainstorming as I write!) Let's try this... expand the user instruction at the bottom of the screen to inform them that they can use the arrow keys to move between records, changing data along the way as they wish. When they press ENTER, have them exit the browse screen (as with the others so far), but keep the screen on the CRT and prompt them whether they are done making changes. If they answer no, then loop them back to the browse screen. I suspect new users will change a number and press ENTER (rather than an arrow), so the prompt will give them a chance to go back to the browse without having to go through another menu. (By the way, I noticed that this particular browse screen does not have append disabled, so the use can "accidentally" add new records. We need to disable this feature.)

On to "Add LIN From CASCOM List". Again, if I can provide a "master" LIN list, we can do away with the redundant nomenclature fields. When I select a LIN, I am getting a quantity from somewhere as a default. This needs to be set to zero as the user enters the routine. Once I add a LIN, I get a "different" browse screen with my new LIN added at the bottom. I don't quite understand why the user is placed in a browse screen at this point, but if you feel it is useful, then the appearance of the browse screen should be similar to that used for changing LIN quantities. Also, the list should be indexed so that it appears in LIN sequence, rather than the added LINs appearing at the bottom.

When I use the "Delete LIN" menu option, I get an error message:
        ALIAS not found: CUNIT
        SELECT 14
I can get back into the routine by telling the program to "Ignore" the error, but I don't think it is working right from that point on. I can see your prompt to confirm deletion of a LIN, and this looks good. You may want to show the quantity of the LIN along with the other information. Also, you may want to handle a Yes/No situation with a prompt such as "Delete This LIN (Y/N)?" and force the user to commit one way or another. Then trap the input, check for a Y or an N, and, if neither is pressed, beep and go back to the prompt.

On to the "Delete Unit" option from the "Unit" menu. As before, when presenting the user with a list of units to choose from, be sure to include the SRC number and the strength as well.

Now I'm going to go to the next "major" menu option: "Task Org". First of all, task organizations are not limited to only those units that a user creates. They can create task organizations out of "standard" SRCs as well, or a combination of both. Secondly, when assigning units to a task organization, the user must be able to indicate a quantity associated with each unit. Recommend you change the True/False approach to a quantity field. Look how the job is done in the current SURE program. Last, we want to be able to include a task

57

force within a task force. Therefore, we probably should ask the user to assign a task force a number analogous to an SRC.

On to the "Profiles" menu option. First, I would recommend either using a different term than "profile" or at least modifying the term with the word "operational" in order to distinguish these profiles from equipment usage profiles. Perhaps "parameters" or "operational parameters" would be useful.

As we cycle through the parameters, let's do it in sequence by class of supply, starting with Classes I, II (you need to add this), III, IV (you need to add this one), V, VI, VIII, and water. You may want to make your original entry screen (for creating a new profile) consistent with the screen you use for editing an existing profile. As the user selects the parameters, they should not be shown all the corresponding fields. For example, when selecting rations, just show the ration types; there is no need to show the unit of measure or the weight per ration. Another example would be for Water: ask two questions, one involving organization level (CO, BTN, BDE, DIV, Above DIV) and another about climate (Temperate, Arctic, Tropic, Arid). Then use the program to look up the proper factors in the database. The user should be able to ESC out at any time, and they should be allowed to prepare an "incomplete" set of parameters (i.e., they may not care about anything but Classes III and V, so why force them to define other parameters). Then when the reports are run, and if they select a class of supply for which no parameters have been defined, they will be prompted to complete the parameters at that time.

On to the "Orders" section of the main menu... As I create an order I am asked to provide the number of phases. Then I am allowed to define the phase lengths, but I am permitted to enter phase lengths for a number of phases greater than the number I defined.

As I look through this part of the program, I sense that you're trying to make the order creation process more complex than necessary. I would propose that we define an order as applying to a SINGLE task organization. Then, as one creates an order they are prompted to identify the task organization (or single SRC if they like) to which the order applies. Then as phases are defined, they can assign the mission profiles (operational parameters) at the same time they assign the phase length. If we allow for the flexibility of having multiple task organizations within a single order, and allow for each organization to have its own set of mission profiles, then I think we are opening up a can of worms we'll never get closed again. Next thing the user would want would be variable length phases for each of his task organizations, and separate subtotals on the report for each of the task organizations. Which almost brings us back to what they would get if they defined a set of orders for each separate task organization in the first place! Perhaps, as a flexibility measure, we can allow the user to copy the contents of an existing order into a new one and then modify those parameters which they feel must change.

On to the reports!!! I tried running through the orders-based reports and selected the order that you defined. I don't think the processing is going on quite the way you intended it. It seems that the process works with ALL the task forces in the file as opposed to just those associated with the order selected.

58

With respect to the quick SRC-based reports (which is essentially what the current SURE program is limited to), I envision the user being able to select from a list of units including standard (CASCOM) SRCs as well as those defined by the user as a task organization. Then I would prompt the user to select which classes of supply for which they want reports. Following this would be a query asking for the input of a pre-defined set of parameters or the option to create a set "on the fly". The program would then check the parameters against the selected classes of supply for the report and prompt the user to complete any necessary parameters that have been left out. At this point, the report would be run. Since your existing report menu is essentially a sequence of dependent operations, I would suggest, based on my earlier discussion, that a menu is not the best way to solicit this type of input. Again, menus should be limited to a list of options which are sequentially independent.

I would now like to address the report format. (I haven't had a chance to check the math yet, so I'll save my comments on that until another time.) I'm afraid we must have the reports done in a TABULAR format to the maximum extent possible. Realizing that all the information will not fit on a screen (widthwise), I would suggest creating a temporary dBASE file to hold the report information and let the user "browse" the report that way. Check the current SURE reports and see what you can do to provide your information in a similar format. If we have to insist that a user have an Epson-compatible dot matrix printer (so that the reports can be compressed with a CHR$(15)), then so be it. But it is absolutely necessary to keep the reports in rows and columns format.

The fuel consumption I'm seeing is showing up in whole numbers. We need this shown to two decimal places. Also, we need to see the total weight and total cube associated with each LIN. I noted that you provide a summary statistic for fuel tank capacity. You may want to include this on a line-by-line basis as well. (I'm not certain how useful this would be, but it may be of some importance.) We also need to show Diesel and JP4 converted into JP8 as a summary statistic. The formula is JP8_Weight = Diesel_Weight * .095 + JP4_Weight * 1.05. JP8 Gallons is just the straight sum of diesel and JP4. Same with JP8 Cube.

Similarly, your ammo reports need more information, such as individual DODIC weight and cube (with decimal places). Also, note that there are, in many cases, many DODICs for a given LIN. Your report appears to be showing only one DODIC per LIN. This is a one-to-many relationship. See the current SURE reports.

One feature that the current SURE program has which I don't see in your program is the ability to create/modify equipment usage profiles for use with Class III planning factors.

Also, I note that you appear to be headed in the direction of a "separate" report for population-based planning factors. I know this is the way the current SURE program does it, but this is one area where I would like the new program to differ. As I said earlier, what I envision is the user selecting which classes of supply he or she wishes to process and the program acting accordingly. You can draw the population from the SRC/task force information.

Just got your phone call... so I'll now take a look at the "system reports" that you've defined. The CASCOM SRCs report is a good idea. However, the more that I think about it, the more I think we ought to choose a different term than "CASCOM" SRCs. Perhaps "standard"

59

would be a better word than CASCOM. Using CASCOM implies that we develop the SRCs or have ownership of them. As these reports scroll on the screen, we must make ESC active to back out. (Currently, I think ESC will crash the program.) The SRC with LINs report is nice, but it appears to run very slowly. I note the other reports are not available yet. The printer defaults is an option under both "Reports" and "Utilities". Suggest we put it under just utilities. As we get further along, I may suggest some refinements as to exactly what a user can specify relative to the printer.

That about does it. I know this will be a lot to take in at one time, but I hope my comments will prove useful. As you're working on addressing what I have discussed, I will be getting with the functional experts within the Planning Factors Branch to discuss your previous README files.

Congratulations on the new addition to your family. I look forward to hearing from you soon.

Bob

Minimum Essential Requirements
for Acceptable SURE V 2.0 Completion

1. **SURE V 1.x Functionality.** The new SURE program must at least be able to perform the same functions as the existing program. This includes, but is not limited to, the following:

   A. The ability to modify and edit an SRC (to include SRC number, title, strength, and LIN composition) and save the modified unit as a "new" SRC that is treated as an equivalent to any other SRC.

   B. The ability to build and modify a task force which consists of a series of one to many SRCs along with the quantity of each SRC.

   C. The ability to correctly calculate POL and/or ammunition requirements based on an "on the fly" list of SRCs and quantities, or based on a previously designed task force.

   D. The ability to correctly calculate population-based requirements.

2. **One Pass Operation.** The user should be able to select for which classes of supply they desire a report. The report will then be prepared according to which classes were selected. There should be no need to make separate passes for POL, ammunition and population-based classes of supply.

3. **SRC Number is Key Field.** The use of the unit name as a key field is unacceptable. The SRC number must be established as the key field. As the user builds their own units, they should be asked to assign their own SRC. Proper program checks should be made to preclude duplication of SRC numbers.

4. **Ability to Back Out Using ESC.** The user must be able to back out from virtually all operations by pressing the ESC key. Menus must be constructed to allow a user to return to the previous menu and the ESC key must be made active to perform the same function. Hitting ESC at any browse screen should return the user to the previous level without making a selection.

5. **Reports Must be Accurate, Complete and Tabular and Must be Viewable on the Screen.** The reports must contain the same elements of information as contained in the SURE V 1.x reports. The calculations must be accurate. The numeric information must be presented in a tabular fashion for readability's sake. The user must be able to view the report on the screen similar to the capability found in SURE 1.x.

6. **Debug, Debug, Debug.** While it is impossible to catch every bug in a program the size of SURE, it is imperative that every reasonable attempt be made to ensure that the program is operating according to design (logical bugs) and that it is as crashproof as possible (programming bugs). Many of the bugs pointed out in the comments of 16-17 Feb 94 still exist, plus many more have crept in. Thorough testing of each feature is essential.

Features That Are Highly Desirable
for an Effective SURE V 2.0

1. **Use ENTER to Move Forward.** In lieu of Ctrl-W or other non-intuitive key sequences, use the ENTER key to indicate positive selection and the desire on the part of the user to move on to the next step.

2. **Task Force Within a Task Force.** This feature has been mentioned from the outset of program development. Perhaps the most effective way to accomplish this is to really establish two separate capabilities:

> A. The would be a new capability to build a custom "unit" based on one to many SRCs (with an appropriate quantity of each). This unit would be assigned its own unique SRC number and would be treated analogous to any other SRC. This way, a user could construct a battalion from companies, a brigade from battalions, a division from brigades, etc.

> B. The capability to build a "task force" consisting of one to many SRCs (with an appropriate quantity of each). This capability is already being programmed.

By combining the two capabilities, the user can effectively create a task force within a task force.

3. **Restructure Menus.** The program has gotten so complex that it is difficult for a user to know what to do first when they enter the program. In addition, many of the menu options are sequentially dependent, a feature which is not desired. Sequential steps should be handled through program operations, not menu options. When a user starts SURE, they really only want to do one of two things: run a report or perform some sort of file maintenance. A file maintenance sub-menu could then contain the categories available for modification, such as: units, task forces, supply usage parameters, equipment usage profiles, etc. Any scenario building capability would also logically fall within the file maintenance area.

-2-

Features That Are Desirable, but Not Necessary for an Effective SURE V 2.0

1. **Scenario Building.** While it is recognized that a great deal of programming effort to date has been spent working on this feature, CASCOM views this to be a desirable capability only after the minimum essential capability has been established. It may be necessary to postpone this feature until a future release of SURE.

2. **Context-sensitive Help.** A context-sensitive help system will be of great benefit to any user. However, a well-constructed user interface with descriptive prompts and adequate documentation is considered adequate in the near term. Only after minimum essential and highly desirable features are implemented should time of any consequence be spent on implementing context-sensitive help.

ANNEX D

THE SURE 2.0 PROGRAM

Table of Contents
Source Code Printout
for SURE 2.0

```
*:********************************************************************
*:
*:          Program: D:\SURE\BACKDBF.PRG
*:
*:          Called by: UTIL_PRO                (procedure in SURE.PRG)
*:
*:              Uses: CSRCLIST.DBF
*:                  : &FILEVAR.DBF
*:                  : CSRCLINQ.DBF
*:                  : TASKORG.DBF
*:                  : ORDERTAS.DBF
*:                  : PROFILES.DBF
*:                  : INTENSTY.DBF
*:                  : CPROFILE.DBF
*:                  : ORDERLST.DBF
*:
*:          MDX files: CSRCLIST.MDX
*:                   : CSRCLINQ.MDX
*:                   : TASKORG.MDX
*:                   : ORDERTAS.MDX
*:                   : PROFILES.MDX
*:                   : INTENSTY.MDX
*:                   : CPROFILE.MDX
*:                   : ORDERLST.MDX
*:                   : &FILEVAR
*:
*:          Documented 04/16/94 at 11:02              SNAP!  version 5.00
*:********************************************************************
STORE " " TO diskvar
STORE HOME() TO homevar
ACTIVATE WINDOW small
@ 1,1 SAY "Enter drive to backup to"
@ 2,2 SAY "or Return to Quit: "
@ ROW(), COL() GET diskvar
READ
IF diskvar=" "
   DEACTIVATE WINDOW small
   RETURN
ENDIF (diskvar=" ")
CLEAR
@ 1,2 SAY "Please wait, backing up..."
diskvar=diskvar+":\"
filevar=diskvar+"CSRCLIST.DBF"
@ 3,2 SAY "                    "
@ 3,2 SAY filevar
SET MESSAGE TO "Backup to Floppy Disk"
COPY FILE csrclist.dbf TO &filevar
filevar=diskvar+"CSRCLINQ.DBF"
@ 3,2 SAY "                    "
@ 3,2 SAY filevar
COPY FILE csrclinq.dbf TO &filevar
filevar=diskvar+"TASKORG.DBF"
@ 3,2 SAY "                    "
@ 3,2 SAY filevar
COPY FILE taskorg.dbf TO &filevar
filevar=diskvar+"ORDERTAS.DBF"
@ 3,2 SAY "                    "
@ 3,2 SAY filevar
COPY FILE ordertas.dbf TO &filevar
filevar=diskvar+"PROFILES.DBF"
```

67

```
@ 3,2 SAY "                    "
@ 3,2 SAY filevar
COPY FILE profiles.dbf TO &filevar
filevar=diskvar+"INTENSTY.DBF"
@ 3,2 SAY "                    "
@ 3,2 SAY filevar
COPY FILE intensty.dbf TO &filevar
filevar=diskvar+"CPROFILE.DBF"
@ 3,2 SAY "                    "
@ 3,2 SAY filevar
COPY FILE cprofile.dbf TO &filevar
filevar=diskvar+"ORDERLST.DBF"
@ 3,2 SAY "                    "
@ 3,2 SAY filevar
COPY FILE orderlst.dbf TO &filevar
filevar=diskvar+"CSRCLIST.MDX"
@ 3,2 SAY "                    "
@ 3,2 SAY filevar
COPY FILE csrclist.mdx TO &filevar
filevar=diskvar+"CSRCLINQ.MDX"
@ 3,2 SAY "                    "
@ 3,2 SAY filevar
COPY FILE csrclinq.mdx TO &filevar
filevar=diskvar+"TASKORG.MDX"
@ 3,2 SAY "                    "
@ 3,2 SAY filevar
COPY FILE taskorg.mdx TO &filevar
filevar=diskvar+"ORDERTAS.MDX"
@ 3,2 SAY "                    "
@ 3,2 SAY filevar
COPY FILE ordertas.mdx TO &filevar
filevar=diskvar+"PROFILES.MDX"
@ 3,2 SAY "                    "
@ 3,2 SAY filevar
COPY FILE profiles.mdx TO &filevar
filevar=diskvar+"INTENSTY.MDX"
@ 3,2 SAY "                    "
@ 3,2 SAY filevar
COPY FILE cprofile.mdx TO &filevar
filevar=diskvar+"ORDERLST.MDX"
@ 3,2 SAY "                    "
@ 3,2 SAY filevar
COPY FILE orderlst.mdx TO &filevar
DEACTIVATE WINDOW small
RETURN

*: EOF: BACKDBF.PRG
```

```
*:******************************************************************
*:
*:  Procedure file: D:\SURE\CALCAMMO.PRG
*:
*:          Set by: GETORDER              (procedure in EDITORDE.PRG)
*:
*:           Calls: STARTHLP.PRG
*:               : AMMOCALC              (procedure in PRINTAMM.PRG)
*:
*:            Uses: TASKORG.DBF
*:               : CSRCLIST.DBF
*:               : ORDERLST.DBF
*:               : CPROFILE.DBF
*:               : PROFILES.DBF
*:               : AMMORATE.DBF
*:               : POSTURE.DBF
*:               : INTENSTY.DBF
*:               : CSRCLINQ.DBF
*:               : ORDERTAS.DBF
*:               : DODIC.DBF
*:               : AMMOEST.DBF
*:
*:       MDX files: TASKORG.MDX
*:               : CSRCLIST.MDX
*:               : ORDERLST.MDX
*:               : CPROFILE.MDX
*:               : PROFILES.MDX
*:               : AMMORATE.MDX
*:               : INTENSTY.MDX
*:               : CSRCLINQ.MDX
*:               : ORDERTAS.MDX
*:               : DODIC.MDX
*:               : AMMOEST.MDX
*:
*:       Documented 04/16/94 at 11:02           SNAP!  version 5.00
*:******************************************************************
* OPEN FILES
CLOSE DATA
STORE PROGRAM() TO progvar
DO starthlp
SELECT 1
USE taskorg
SELECT 2
USE csrclist
SELECT 3
USE orderlst
SELECT 4
USE cprofile
SELECT 7
USE profiles
SELECT 8
USE ammorate ORDER lin
SELECT 17
USE posture
SELECT 10
USE intensty
SELECT 14
USE csrclinq
SELECT 18
USE ordertas
```

69

```
SELECT 19
USE dodic ORDER dodic
SELECT 21
USE ammoest
* SET RELATIONS
SELECT 19
SET ORDER TO dodic
SELECT 8
SET ORDER TO dodic
SET RELATION TO dodic INTO dodic
SELECT 8
SET ORDER TO lin
*SELECT 14
*SET ORDER TO lin
*SET RELATION TO lin INTO ammorate
SELECT 14
SET ORDER TO unitname
SELECT 2
SET ORDER TO unitname
SELECT 1
SET ORDER TO unitname
SET RELATION TO unitname INTO csrclinq, unitname INTO csrclist
SELECT 7
SET ORDER TO profile
SELECT 4
SET ORDER TO profile
SET RELATION TO profile INTO profiles
SELECT 4
SET ORDER TO profilenam
SELECT 18
SET ORDER TO profilenam
SET RELATION TO profilenam INTO cprofile
SELECT 3
SET ORDER TO NUMBER
SEEK numberva
SELECT 21
ZAP
* Procedure to calculate AMMO consumption
* Restrict ordertas.dbf to records in this Order
SELECT 18
* Put in taskorgnam+str(phase) order
SET FILTER TO
SET FILTER TO NUMBER=orderlst->number
GO TOP
SET ORDER TO taskphase
GO TOP
DO WHILE .NOT. EOF()
    * only work with records in this taskorder
    SELECT 1
    SET FILTER TO
    SET FILTER TO taskorg->taskorgnam=ordertas->taskorgnam
    GO TOP
    ACTIVATE SCREEN
    @ 17,2 CLEAR TO 17,78
    @ 17,2 SAY taskorgnam
    ACTIVATE WINDOW process
    * Do one unit at a time
    DO WHILE .NOT. EOF()
        * Go LIN by LIN for this unit
        SELECT 14
```

70

```
        SET FILTER TO
        SET FILTER TO csrclinq->unitname=taskorg->unitname
        GO TOP
        DO ammocalc
        SELECT 1
        SKIP
        IF EOF()
            EXIT
        ENDIF (EOF())
    ENDDO (.NOT. EOF())
    SELECT 18
    SKIP
    IF EOF()
        EXIT
    ENDIF (EOF())
ENDDO (.NOT. EOF())
*!*****************************************************************
*!
*!      Procedure: AMMOCALC
*!
*!      Called by: CALCAMMO.PRG
*!               : PRIN_PRO              (procedure in SYSREP.PRG)
*!
*!*****************************************************************
PROCEDURE ammocalc
ACTIVATE SCREEN
@ 18,2 CLEAR TO 18,78
@ 18,2 SAY unitname
ACTIVATE WINDOW process
DO WHILE .NOT. EOF()
    ammovar=0
    * If ammo user, calculate ammo
    SELECT 8
    SEEK csrclinq->lin
    IF FOUND()
        DO WHILE ammorate->lin=csrclinq->lin
            *IF ammorate->dodic<>" "
            SELECT 21
            APPEND BLANK
            ? cprofile->posture
            DO CASE
            CASE cprofile->posture="Defense First Day"
                ammovar=csrclinq->quantity*ammorate->hvyld
            CASE cprofile->posture="Defense Succeeding Days"
                ammovar=csrclinq->quantity*ammorate->hvysd
            CASE cprofile->posture="Attack First Day"
                ammovar=csrclinq->quantity*ammorate->hvyla
            CASE cprofile->posture="Attack Succeeding Days"
                ammovar=csrclinq->quantity*ammorate->hvysa
            CASE cprofile->posture="Attack Protracted Period"
                ammovar=csrclinq->quantity*ammorate->hvyp
            ENDCASE
            * Multiply ammo use by intensity factor
            ammovar=ammovar*cprofile->factor
            * Multiply ammo use by strength in this phase
            ammovar=ammovar*ordertas->strength/100
            * Multiply ammo use by length of this phase
            * first we need to know what phase we're in
            phasevar="PHASE"+LTRIM(STR(ordertas->phase))+"LEN"
            * then multiply by days in this phase
```

71

```
        ACTIVATE SCREEN
        @ 16,26 CLEAR TO 16,36
        @ 16,26 SAY "Phase "+LTRIM(STR(ordertas->phase))
        ACTIVATE WINDOW process
        DO CASE
        CASE phasevar="PHASE1LEN"
            ammovar=ammovar*orderlst->phase1len
        CASE phasevar="PHASE2LEN"
            ammovar=ammovar*orderlst->phase2len
        CASE phasevar="PHASE3LEN"
            ammovar=ammovar*orderlst->phase3len
        CASE phasevar="PHASE4LEN"
            ammovar=ammovar*orderlst->phase4len
        CASE phasevar="PHASE5LEN"
            ammovar=ammovar*orderlst->phase5len
        CASE phasevar="PHASE6LEN"
            ammovar=ammovar*orderlst->phase6len
        CASE phasevar="PHASE7LEN"
            ammovar=ammovar*orderlst->phase7len
        CASE phasevar="PHASE8LEN"
            ammovar=ammovar*orderlst->phase8len
        CASE phasevar="PHASE9LEN"
            ammovar=ammovar*orderlst->phase9len
        ENDCASE
        ? STR(ammovar)
        REPLACE ammoest->calc_qty WITH ammovar
        REPLACE ammoest->dodic WITH ammorate->dodic
        ? ammorate->dodic
        ?? " "+ammorate->dodic_nom
        REPLACE ammoest->cbt_loadqt WITH
csrclinq->quantity*ammorate->cbt_load;
            *ordertas->strength/100
        REPLACE ammoest->number WITH orderlst->number
        REPLACE ammoest->unitname WITH taskorg->unitname
        REPLACE ammoest->src WITH taskorg->src
        REPLACE ammoest->qty WITH taskorg->qty
        REPLACE ammoest->taskorgnam WITH taskorg->taskorgnam
        REPLACE ammoest->phase WITH ordertas->phase
        REPLACE ammoest->lin WITH ammorate->lin
        REPLACE ammoest->lin_nomen WITH ammorate->lin_nomen
        REPLACE ammoest->quantity WITH
csrclinq->quantity*ordertas->strength/100
        SELECT 8
        SKIP
     ENDDO (ammorate->lin=csrclinq->lin)
   ENDIF (FOUND())
   SELECT 14
   SKIP
   IF EOF()
      EXIT
   ENDIF (EOF())
ENDDO (.NOT. EOF())

*: EOF: CALCAMMO.PRG
```

```
*:*****************************************************************
*:
*: Procedure file: D:\SURE\CALCOTHE.PRG
*:
*:  Procs & Fncts: OTHERCAL
*:
*:         Set by: GETORDER              (procedure in EDITORDE.PRG)
*:
*:          Calls: STARTHLP.PRG
*:              : OTHERCAL               (procedure in CALCOTHE.PRG)
*:
*:           Uses: TASKORG.DBF
*:              : CSRCLIST.DBF
*:              : ORDERLST.DBF
*:              : CPROFILE.DBF
*:              : PROFILES.DBF
*:              : POSTURE.DBF
*:              : INTENSTY.DBF
*:              : CSRCLINQ.DBF
*:              : ORDERTAS.DBF
*:              : MEDICAL.DBF
*:              : WATER.DBF
*:              : ECHELONS.DBF
*:              : CL_OTHER.DBF
*:
*:      MDX files: TASKORG.MDX
*:              : CSRCLIST.MDX
*:              : ORDERLST.MDX
*:              : CPROFILE.MDX
*:              : PROFILES.MDX
*:              : INTENSTY.MDX
*:              : CSRCLINQ.MDX
*:              : ORDERTAS.MDX
*:              : MEDICAL.MDX
*:              : WATER.MDX
*:              : ECHELONS.MDX
*:              : CL_OTHER.MDX
*:
*:      Documented 04/16/94 at 11:02                    SNAP!  version 5.00
*:*****************************************************************
* OPEN FILES
CLOSE DATA
STORE PROGRAM() TO progvar
DO starthlp
SELECT 1
USE taskorg
SELECT 2
USE csrclist
SELECT 3
USE orderlst
SELECT 4
USE cprofile
SELECT 7
USE profiles
SELECT 17
USE posture
SELECT 10
USE intensty
SELECT 14
USE csrclinq
```

73

```
SELECT 18
USE ordertas
SELECT 22
USE medical ORDER med_name
SELECT 23
USE water ORDER water_name
SELECT 25
USE echelons ORDER echelon
SELECT 26
USE cl_other
* SET RELATIONS
SELECT 2
SET ORDER TO echelon
SET RELATION TO echelon INTO echelons
SELECT 14
SET ORDER TO unitname
SELECT 2
SET ORDER TO unitname
SELECT 1
SET ORDER TO unitname
SET RELATION TO unitname INTO csrclinq, unitname INTO csrclist
SELECT 7
SET ORDER TO profile
SELECT 4
SET ORDER TO water_name
SET RELATION TO water_name INTO water
SET ORDER TO med_name
SET RELATION TO med_name INTO medical
SET ORDER TO profile
SET RELATION TO profile INTO profiles
SELECT 4
SET ORDER TO profilenam
SELECT 18
SET ORDER TO profilenam
SET RELATION TO profilenam INTO cprofile
SELECT 3
SET ORDER TO NUMBER
SEEK numberva
* Calculate CL I, H2O, CL IV, CL VIII
SELECT 26
ZAP
* Restrict ordertas.dbf to records in this Order
SELECT 18
SET FILTER TO
SET FILTER TO NUMBER=orderlst->number
GO TOP
* Put in taskorgnam+str(phase) order
SET ORDER TO taskphase
GO TOP
DO WHILE .NOT. EOF()
    * only work with records in this taskorder
    SELECT 1
    SET FILTER TO
    SET FILTER TO taskorg->taskorgnam=ordertas->taskorgnam
    GO TOP
    ACTIVATE SCREEN
    @ 17,2 CLEAR TO 17,78
    @ 17,2 SAY taskorgnam
    ACTIVATE WINDOW process
    * Do one unit at a time
```

74

```
      DO WHILE .NOT. EOF()
         * Go LIN by LIN for this unit
         SELECT 14
         SET FILTER TO
         SET FILTER TO csrclinq->unitname=taskorg->unitname
         GO TOP
         DO othercal
         SELECT 1
         namevar=unitname
         DO WHILE .T.
            SKIP
            IF EOF() .OR. namevar<>unitname
               EXIT
            ENDIF (EOF() .OR. namevar<>unitname)
         ENDDO (.T.)
         IF EOF()
            EXIT
         ENDIF (EOF())
      ENDDO (.NOT. EOF())
      SELECT 18
      SKIP
      IF EOF()
         EXIT
      ENDIF (EOF())
ENDDO (.NOT. EOF())
*!*****************************************************************
*!
*!        Procedure: OTHERCAL
*!
*!        Called by: CALCOTHE.PRG
*!
*!*****************************************************************
PROCEDURE othercal
ACTIVATE SCREEN
@ 18,2 CLEAR TO 18,78
@ 18,2 SAY unitname
ACTIVATE WINDOW process
phaselnv=0
SELECT 26
APPEND BLANK
REPLACE cl_other->number WITH orderlst->number
REPLACE cl_other->unitname WITH taskorg->unitname
REPLACE cl_other->src WITH taskorg->src
REPLACE cl_other->qty WITH taskorg->qty
REPLACE cl_other->taskorgnam WITH taskorg->taskorgnam
REPLACE cl_other->phase WITH ordertas->phase
REPLACE cl_other->profilenam WITH ordertas->profilenam
* Multiply ammo use by length of this phase
* first we need to know what phase we're in
phasevar="PHASE"+LTRIM(STR(ordertas->phase))+"LEN"
* then multiply by days in this phase
ACTIVATE SCREEN
@ 16,26 CLEAR TO 16,36
@ 16,26 SAY "Phase "+LTRIM(STR(ordertas->phase))
ACTIVATE WINDOW process
DO CASE
CASE phasevar="PHASE1LEN"
   phaselnv=orderlst->phase1len
CASE phasevar="PHASE2LEN"
   phaselnv=orderlst->phase2len
```

75

```
CASE phasevar="PHASE3LEN"
    phaselnv=orderlst->phase3len
CASE phasevar="PHASE4LEN"
    phaselnv=orderlst->phase4len
CASE phasevar="PHASE5LEN"
    phaselnv=orderlst->phase5len
CASE phasevar="PHASE6LEN"
    phaselnv=orderlst->phase6len
CASE phasevar="PHASE7LEN"
    phaselnv=orderlst->phase7len
CASE phasevar="PHASE8LEN"
    phaselnv=orderlst->phase8len
CASE phasevar="PHASE9LEN"
    phaselnv=orderlst->phase9len
ENDCASE
ACTIVATE WINDOW process
* Calculate Class I
* Strenth for Rations
*    Numeric strength
REPLACE cl_other->strength WITH csrclist->strength*ordertas->strength/100
? "Numeric Strength: "+STR(cl_other->strength)
* Class I
REPLACE cl_other->cl_i_qty WITH
csrclist->strength*phaselnv*ordertas->strength/100
* Gallons of water
REPLACE cl_other->h2o_qty WITH csrclist->strength*phaselnv*water->gal_man_da*;
    ordertas->strength/100
? "Gallons H2O: "+STR(cl_other->h2o_qty)
* Class IV barrier
REPLACE cl_other->cl_ivbqty WITH
csrclist->strength*phaselnv*4*ordertas->strength/100
? "Barrier Material: "+STR(cl_other->cl_ivbqty)
REPLACE cl_other->cl_ivcqty WITH
csrclist->strength*phaselnv*4.5*ordertas->strength/100
? "Construction Material: "+STR(cl_other->cl_ivcqty)
REPLACE cl_other->cl_viii_qt WITH
csrclist->strength*phaselnv*medical->plan_fact
? "Class VIII: "+STR(cl_other->cl_viii_qt)
ACTIVATE SCREEN

*: EOF: CALCOTHE.PRG
```

```
*:**********************************************************************
*:
*: Procedure file: D:\SURE\CALCPOL.PRG
*:
*:            Set by: GETORDER             (procedure in EDITORDE.PRG)
*:
*:            Calls: STARTHLP.PRG
*:                 : POLCALC              (procedure in PRINTPOL.PRG)
*:
*:            Uses: TASKORG.DBF
*:                : CSRCLIST.DBF
*:                : ORDERLST.DBF
*:                : CPROFILE.DBF
*:                : PROFILES.DBF
*:                : POLRATE.DBF
*:                : POSTURE.DBF
*:                : INTENSTY.DBF
*:                : POLWGT.DBF
*:                : CSRCLINQ.DBF
*:                : ORDERTAS.DBF
*:                : POLEST.DBF
*:
*:        MDX files: TASKORG.MDX
*:                 : CSRCLIST.MDX
*:                 : ORDERLST.MDX
*:                 : CPROFILE.MDX
*:                 : PROFILES.MDX
*:                 : POLRATE.MDX
*:                 : INTENSTY.MDX
*:                 : POLWGT.MDX
*:                 : CSRCLINQ.MDX
*:                 : ORDERTAS.MDX
*:                 : POLEST.MDX
*:
*:        Documented 04/16/94 at 11:02              SNAP!  version 5.00
*:**********************************************************************
* OPEN FILES
CLOSE DATA
DO starthlp
SELECT 1
USE taskorg
SELECT 2
USE csrclist
SELECT 3
USE orderlst
SELECT 4
USE cprofile
SELECT 7
USE profiles
SELECT 9
USE polrate
SELECT 17
USE posture
SELECT 10
USE intensty
SELECT 12
USE polwgt ORDER pol_type
SELECT 14
USE csrclinq
SELECT 18
```

```
USE ordertas
SELECT 20
USE polest
* SET RELATIONS
SELECT 12
SET ORDER TO pol_type
SELECT 9
SET ORDER TO pol_type
SET RELATION TO pol_type INTO polwgt
SELECT 9
SET ORDER TO lin
SELECT 14
SET ORDER TO lin
SET RELATION TO lin INTO polrate
SELECT 14
SET ORDER TO unitname
SELECT 2
SET ORDER TO unitname
SELECT 1
SET ORDER TO unitname
SET RELATION TO unitname INTO csrclinq, unitname INTO csrclist
SELECT 7
SET ORDER TO profile
SELECT 4
SET ORDER TO profile
SET RELATION TO profile INTO profiles
SELECT 4
SET ORDER TO profilenam
SELECT 18
SET ORDER TO profilenam
SET RELATION TO profilenam INTO cprofile
SELECT 12
SET ORDER TO pol_type
SELECT 9
SET ORDER TO pol_type
SET RELATION TO pol_type INTO polwgt
SET ORDER TO lin
SELECT 3
SET ORDER TO NUMBER
SEEK numberva
SELECT 20
ZAP
* Procedure to calculate POL consumption
* Restrict ordertas.dbf to records in this Order
SELECT 18
SET FILTER TO
SET FILTER TO NUMBER=orderlst->number
GO TOP
* Put in taskorgnam+str(phase) order
SET ORDER TO taskphase
GO TOP
DO WHILE .NOT. EOF()
    * only work with records in this taskorder
    SELECT 1
    SET FILTER TO
    SET FILTER TO taskorg->taskorgnam=ordertas->taskorgnam
    GO TOP
    ACTIVATE SCREEN
    @ 17,2 CLEAR TO 17,78
    @ 17,2 SAY taskorgnam
```

78

```
      ACTIVATE WINDOW process
      * Do one unit at a time
      DO WHILE .NOT. EOF()
         * Go LIN by LIN for this unit
         SELECT 14
         SET FILTER TO
         SET FILTER TO csrclinq->unitname=taskorg->unitname
         GO TOP
         DO polcalc
         SELECT 1
         SKIP
         IF EOF()
            EXIT
         ENDIF (EOF())
      ENDDO (.NOT. EOF())
      SELECT 18
      SKIP
      IF EOF()
         EXIT
      ENDIF (EOF())
   ENDDO (.NOT. EOF())
   *!********************************************************************
   *!                       '
   *!        Procedure: POLCALC
   *!
   *!        Called by: CALCPOL.PRG
   *!                 : PRIN_PRO                 (procedure in SYSREP.PRG)
   *!
   *!********************************************************************
   PROCEDURE polcalc
   ACTIVATE SCREEN
   @ 18,2 CLEAR TO 18,78
   @ 18,2 SAY unitname
   ACTIVATE WINDOW process
   DO WHILE .NOT. EOF()
      * If fuel user, calculate fuel
      IF polrate->equip_cat<>" "
         ? polrate->lin
         ? polrate->lin_nomen
         ? polrate->equip_cat
         polvar=0
         SELECT 20
         APPEND BLANK
         DO CASE
         CASE polrate->equip_cat="AE"
            polvar=csrclinq->quantity*polrate->idle_avg*profiles->ae_hrs
         CASE polrate->equip_cat="AV"
            polvar=csrclinq->quantity*polrate->idle_avg*profiles->av_hrs
            ? csrclinq->quantity
            ? polrate->idle_avg
            ? profiles->av_hrs
         CASE polrate->equip_cat="CE"
            polvar=csrclinq->quantity*polrate->idle_avg*profiles->ce_hrs
         CASE polrate->equip_cat="GN"
            polvar=csrclinq->quantity*polrate->idle_avg*profiles->gn_hrs
         CASE polrate->equip_cat="HG"
            polvar=csrclinq->quantity*polrate->idle_avg*profiles->hg_hrs
         CASE polrate->equip_cat="MH"
            polvar=csrclinq->quantity*polrate->idle_avg*profiles->mh_hrs
         CASE polrate->equip_cat="OV"
```

```
          polvar=csrclinq->quantity*polrate->idle_avg*profiles->ov_hrs
    CASE polrate->equip_cat="SG"
          polvar=csrclinq->quantity*polrate->idle_avg*profiles->sg_hrs
    CASE polrate->equip_cat="SV"
          polvar=csrclinq->quantity*polrate->idle_avg*profiles->sv_hrs
    CASE polrate->equip_cat="WV"
          polvar=csrclinq->quantity*polrate->idle_avg*profiles->wv_kms
    CASE polrate->equip_cat="TV"
          polvar=csrclinq->quantity*polrate->idle_avg*profiles->tv_ti_hrs
          polvar=polvar+csrclinq->quantity*polrate->cc*profiles->tv_cc_hrs
          polvar=polvar+csrclinq->quantity*polrate->sr*profiles->tv_sr_hrs
    ENDCASE
    * multiply pol consumption by phase length
    * first we need to know what phase we're in
    phasevar="PHASE"+LTRIM(STR(ordertas->phase))+"LEN"
    * then multiply by days in this phase
    ACTIVATE SCREEN
    @ 16,26 CLEAR TO 16,36
    @ 16,26 SAY "Phase "+LTRIM(STR(ordertas->phase))
    ACTIVATE WINDOW process
    DO CASE
    CASE phasevar="PHASE1LEN"
          polvar=polvar*orderlst->phase1len
    CASE phasevar="PHASE2LEN"
          polvar=polvar*orderlst->phase2len
    CASE phasevar="PHASE3LEN"
          polvar=polvar*orderlst->phase3len
    CASE phasevar="PHASE4LEN"
          polvar=polvar*orderlst->phase4len
    CASE phasevar="PHASE5LEN"
          polvar=polvar*orderlst->phase5len
    CASE phasevar="PHASE6LEN"
          polvar=polvar*orderlst->phase6len
    CASE phasevar="PHASE7LEN"
          polvar=polvar*orderlst->phase7len
    CASE phasevar="PHASE8LEN"
          polvar=polvar*orderlst->phase8len
    CASE phasevar="PHASE9LEN"
          polvar=polvar*orderlst->phase9len
    ENDCASE
    * now multiply by strength of this task force this phase
    polvar=polvar*ordertas->strength/100
    * put relavent data in polest database
    REPLACE polest->pol_type WITH polrate->pol_type
    REPLACE polest->number WITH orderlst->number
    REPLACE polest->unitname WITH taskorg->unitname
    REPLACE polest->src WITH taskorg->src
    REPLACE polest->qty WITH taskorg->qty
    REPLACE polest->taskorgnam WITH taskorg->taskorgnam
    REPLACE polest->phase WITH ordertas->phase
    REPLACE polest->lin WITH polrate->lin
    REPLACE polest->equip_cat WITH polrate->equip_cat
    REPLACE polest->lbs_gal WITH polwgt->lbs_gal
    REPLACE polest->lin_nomen WITH polrate->lin_nomen
    * Calculate fuel tank capacity for this lin at this phase and strength
    REPLACE polest->ft_cap WITH
polrate->ft_cap*csrclinq->quantity*ordertas->strength/100
    REPLACE polest->quantity WITH csrclinq->quantity*ordertas->strength/100
    ? STR(polvar)
    REPLACE polest->pol_qty WITH polvar
```

80

```
    ENDIF (polrate->equip_cat<>" ")
    SELECT 14
    SKIP
    IF EOF()
        EXIT
    ENDIF (EOF())
ENDDO (.NOT. EOF())

*: EOF: CALCPOL.PRG
```

```
*:********************************************************************
*:                                                              '
*:          Program: D:\SURE\CLOSEALL.PRG
*:
*:          Called by: QUIT_PRO              (procedure in SURE.PRG)
*:
*:          Documented 04/16/94 at 11:02                SNAP!  version 5.00
*:********************************************************************
*closeall.prg
DEACTIVATE MENU
CLOSE ALL
CLEAR
QUIT
*: EOF: CLOSEALL.PRG
```

```
*:********************************************************************
*:
*:          Program: D:\SURE\COMPRESS.PRG
*:
*:          Called by: UTIL_PRO              (procedure in SURE.PRG)
*:
*:     Memory Files: VARIABLE.MEM
*:
*:          Documented 04/16/94 at 11:02              SNAP!  version 5.00
*:********************************************************************
STORE SPACE(10) TO vprintva
ACTIVATE WINDOW small
@ 1,1 SAY "Select printer type with SPACE"
@ 2,1 SAY "and RETURN to select."
@ 3,1 GET vprintva;
    PICTURE "@M Dot Matrix,Laser,DeskJet,Do not set!";
    MESSAGE "Dot Matrix for Epson/IBM compatibles, Laser for HP Laser Printers"
READ
DO CASE
CASE vprintva="Dot Matrix"
    ??? CHR(27)+CHR(15)
CASE vprintva="Laser"
    ??? CHR(27)+"@10O"
    ??? CHR(27)+"(8U"
    ??? CHR(27)+"(s0p16.66h8.5v0s0b0T"
    _PEJECT="AFTER"
CASE vprintva="DeskJet"
    ??? CHR(27)+"(s16.67h12V"
ENDCASE
SAVE TO variable ALL LIKE v*
DEACTIVATE WINDOW small
RETURN

*: EOF: COMPRESS.PRG
```

```
*:****************************************************************
*:                              ,
*:  Procedure file: D:\SURE\CREATEOR.PRG
*:
*:    Procs & Fncts: STARTOR
*:                 : TOMP_PRO
*:                 : ORDS_PRO
*:                 : CREATEO
*:                 : VIEWORDE
*:                 : ASSIGNTA
*:                 : ENDORDR
*:                 : DISPORD
*:                 : DISPORD2
*:                 : 1PROALL
*:                 : 1PROONE
*:                 : 1PRO1PRI
*:                 : PHASEBY
*:                 : VIEWTOP
*:                 : FINDPRO
*:                 : DISPTAS
*:
*:          Set by: ORDE_PRO              (procedure in SURE.PRG)
*:
*:          Calls: STARTOR               (procedure in CREATEOR.PRG)
*:               : TOMP_PRO              (procedure in CREATEOR.PRG)
*:               : ORDS_PRO              (procedure in CREATEOR.PRG)
*:
*:        Documented 04/16/94 at 11:01            SNAP!  version 5.00
*:****************************************************************
* createor.prg
STORE PROGRAM() TO progvar
STORE 0 TO numberva
STORE SPACE(8) TO pnumberv
STORE SPACE(50) TO namevar
STORE 1 TO statusvar
STORE 0 TO nextpop
SET PROCEDURE TO instorde
DO startor
* Popup for Mission Profile/Task Organization assignment choices
DEFINE POPUP tomp_pop FROM 1,3
DEFINE BAR 1 OF tomp_pop PROMPT ;
   "Assign one Parameter Set to all Task Organizations and Phases"
DEFINE BAR 2 OF tomp_pop PROMPT ;
   "Assign a Parameter Set to one Task Organization for all Phases"
DEFINE BAR 3 OF tomp_pop PROMPT ;
   "Assign a Parameter Set to all Task Organizations for a single Phase"
DEFINE BAR 4 OF tomp_pop PROMPT ;
   "Assign Parameter Sets to a Task Organization Phase by Phase"
DEFINE BAR 5 OF tomp_pop PROMPT ;
   "View this Orders Task Organizations and Parameter Sets by Phase"
DEFINE BAR 6 OF tomp_pop PROMPT "Return to Previous Menu"
ON SELECTION POPUP tomp_pop DO tomp_pro
ON BAR 1 OF tomp_pop DO tomp_bar1
ON BAR 2 OF tomp_pop DO tomp_bar2
ON BAR 3 OF tomp_pop DO tomp_bar3
ON BAR 4 OF tomp_pop DO tomp_bar4
ON BAR 5 OF tomp_pop DO tomp_bar5
ON BAR 6 OF tomp_pop DO tomp_bar6
DEFINE POPUP ords_pop FROM 1,23
DEFINE BAR 1 OF ords_pop PROMPT "Create Order"
```

```
DEFINE BAR 2 OF ords_pop PROMPT "View Orders"
DEFINE BAR 3 OF ords_pop PROMPT "Assign Task Organizations"
DEFINE BAR 4 OF ords_pop PROMPT "Assign Parameter Sets to Task Organization
Phases"
DEFINE BAR 5 OF ords_pop PROMPT "Return to Main Menu"
ON SELECTION POPUP ords_pop DO ords_pro
ON BAR 1 OF ords_pop DO ords_bar1
ON BAR 2 OF ords_pop DO ords_bar2
ON BAR 3 OF ords_pop DO ords_bar3
ON BAR 4 OF ords_pop DO ords_bar4
ON BAR 5 OF ords_pop DO ords_bar5
DO WHILE PROMPT()<>"Return to Main Menu"
   ACTIVATE POPUP ords_pop
   IF nextpop=1
      ACTIVATE POPUP tomp_pop
      nexttpop=0
      ACTIVATE POPUP ords_pop
   ENDIF (nextpop=1)
ENDDO (PROMPT()<>"Return to Main Menu")
*!************************************************************************
*!
*!        Procedure: ORDS_PRO
*!
*!        Called by: CREATEOR.PRG
*!
*!           Calls: STARTOR            (procedure in CREATEOR.PRG)
*!                 : CREATEO            (procedure in CREATEOR.PRG)
*!                 : VIEWORDE           (procedure in CREATEOR.PRG)
*!                 : ASSIGNTA           (procedure in CREATEOR.PRG)
*!                 : ENDORDR            (procedure in CREATEOR.PRG)
*!
*!************************************************************************
PROCEDURE ords_pro
DO CASE
CASE BAR()=1
   IF statusvar=2
      DO startor
   ENDIF (statusvar=2)
   DO createo
CASE BAR()=2
   DO vieworde
CASE BAR()=3
   DO assignta
CASE BAR()=4
   IF statusvar=1
      ACTIVATE WINDOW small
      ? CHR(7)
      @ 1,2 SAY "You must create an Order first!"
      @ 3,3 SAY "Press any key..."
      WAIT ""
      DEACTIVATE WINDOW small
      CLEAR
   ELSE
      nextpop=1
      DEACTIVATE POPUP
   ENDIF (statusvar=1)
CASE BAR()=5
   DO endordr
CASE BAR()=6
   DO startor
```

```
ENDCASE
* CASE BAR()=1
CLEAR
*!*******************************************************************
*!
*!        Procedure: CREATEO
*!
*!        Called by: ORDS_PRO           (procedure in CREATEOR.PRG)
*!
*!           Calls: DISPORD            (procedure in CREATEOR.PRG)
*!
*!            Uses: TEMP.DBF
*!
*!*******************************************************************
PROCEDURE createo
SELECT 3
SET ORDER TO NUMBER
COPY STRUCTURE TO temp
SELECT 11
USE temp
APPEND BLANK
DO WHILE .T.
   SET CONFIRM ON
   ACTIVATE WINDOW editor
   @ 1,1  SAY "Order No. "
   @ 1,13 GET NUMBER PICTURE "XXXXXXXX"
   @ 1,22 SAY "Order number cannot be changed later!"
   @ 3,1  SAY "Order Name: "
   @ 3,13 GET name PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
   @ 5,1  SAY "Description:"
   @ 6,6  GET descriptio PICTURE
"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
   @ 8,1  SAY "Number of Phases: "
   @ 8,19 GET phases PICTURE "9"
   @ 10,1 SAY "Phase Lengths (Days):"
   @ ROW(),COL()+3 SAY "P-1"
   @ ROW(),COL()+1 GET phase1len
   @ ROW(),COL()+3 SAY "P-2"
   @ ROW(),COL()+1 GET phase2len
   @ ROW(),COL()+3 SAY "P-3"
   @ ROW(),COL()+1 GET phase3len
   @ ROW(),COL()+3 SAY "P-4"
   @ ROW(),COL()+1 GET phase4len
   @ 12,1
   @ ROW(),COL() SAY "P-5"
   @ ROW(),COL()+1 GET phase5len
   @ ROW(),COL()+3 SAY "P-6"
   @ ROW(),COL()+1 GET phase6len
   @ ROW(),COL()+3 SAY "P-7"
   @ ROW(),COL()+1 GET phase7len
   @ ROW(),COL()+3 SAY "P-8"
   @ ROW(),COL()+1 GET phase8len
   @ ROW(),COL()+3 SAY "P-9"
   @ ROW(),COL()+1 GET phase9len
   @14,2 SAY "No quantity calcuations for Phase Lengths of Zero. Ctrl-W when
Done"
   READ
   SET CONFIRM OFF
   DEACTIVATE WINDOW editor
   IF NUMBER=" "
```

```
            ACTIVATE WINDOW small
            ? CHR(7)
            @ 1,2 SAY "No Order Number!"
            @ 2,2 SAY "Order NOT Added!"
            @ 3,3 SAY "Press any key..."
            WAIT ""
            DEACTIVATE WINDOW small
            DELETE
            PACK
            SELECT 11
            USE
            CLEAR
            EXIT
        ENDIF (NUMBER=" ")
        SELECT 3
        SEEK temp->number
        IF FOUND()
            CLEAR
            SELECT 11
            ACTIVATE WINDOW small
            ? CHR(7)
            CLEAR
            STORE "YES" TO yesnovar
            @ 1,2 SAY "Order Number Duplicate! Type"
            @ 2,2 SAY "YES to try again... "
            @ ROW(),COL() GET yesnovar
            READ
            IF yesnovar="YES"
                DEACTIVATE WINDOW small
            ELSE
                DEACTIVATE WINDOW small
                DELETE
                PACK
                SELECT 11
                USE
                CLEAR
                EXIT
            ENDIF (yesnovar="YES")
        ELSE
            SELECT 11
            USE
            SELECT 3
            APPEND FROM temp
            SELECT 11
            USE temp
            ZAP
            SELECT 11
            USE
            SELECT 3
            namevar=orderlst->name
            pnumberv=orderlst->number
            EXIT
        ENDIF (FOUND())
ENDDO (.T.)
statusvar=2
DO dispord
KEYBOARD "{DNARROW}"
KEYBOARD "{DNARROW}"
* CASE BAR()=2
*!**************************************************************
```

```
*!
*!         Procedure: VIEWORDE
*!
*!         Called by: ORDS_PRO          (procedure in CREATEOR.PRG)
*!                  : ORDT_PRO          (procedure in EDITORDE.PRG)
*!                  : GETORDER          (procedure in EDITORDE.PRG)
*!
*!             Calls: SETBROWS.PRG
*!                  : ENDBROWS.PRG
*!                  : DISPORD2          (procedure in CREATEOR.PRG)
*!                  : DISPORD3          (procedure in ORDEREP1.PRG)
*!
*!*************************************************************************
PROCEDURE vieworde
SELECT 3
* Make sure there are some records to view
IF RECCOUNT()=0
    ACTIVATE WINDOW small
    ? CHR(7)
    @ 1,2 SAY "You must have at"
    @ 2,2 SAY "least one Order!"
    @ 3,3 SAY "Press any key..."
    WAIT ""
    DEACTIVATE WINDOW small
    CLEAR
    RETURN
ENDIF (RECCOUNT()=0)
ACTIVATE WINDOW browser
GO TOP
*      SET MESSAGE TO "Choose an Order to View"
DO setbrows
BROWSE NOMENU NOEDIT NOAPPEND NODELETE FIELDS NUMBER,name, descriptio FREEZE
name COMPRESS
DO endbrows
DEACTIVATE WINDOW browser
SET CURSOR OFF
CLEAR
DO WHILE LASTKEY()<>23
    DO dispord2
    IF RECCOUNT()=1
        @ 23,30 SAY "Crtl-W when Done"
    ELSE
        @ 23,15 SAY "Page-Up or Page-Down to View, Ctrl-W when Done"
    ENDIF (RECCOUNT()=1)
    IF LASTKEY()=3
        SKIP
        IF EOF()
            GO TOP
        ENDIF (EOF())
    ENDIF (LASTKEY()=3)
    IF LASTKEY()=18
        SKIP -1
        IF BOF()
            GO BOTTOM
        ENDIF (BOF())
    ENDIF (LASTKEY()=18)
    WAIT ""
ENDDO (LASTKEY()<>23)
SET CURSOR ON
SET MESSAGE TO
```

```
* CASE BAR()=3
*!*********************************************************************
*!
*!        Procedure: ASSIGNTA
*!
*!        Called by: ORDS_PRO            (procedure in CREATEOR.PRG)
*!                 : ORDT_PRO            (procedure in EDITORDE.PRG)
*!
*!             Uses: TTEMP.DBF
*!                 : TEMP.DBF
*!
*!*********************************************************************
PROCEDURE assignta
IF statusvar=1
   ACTIVATE WINDOW small
   ? CHR(7)
   @ 1,2 SAY "You must create an Order first!"
   @ 3,3 SAY "Press any key..."
   WAIT ""
   DEACTIVATE WINDOW small
   CLEAR
   RETURN
ENDIF (statusvar=1)
* Select Task Organizations for this Order
* Make sure order has 1 or more phases
SELECT 3
IF phases=0
   ACTIVATE WINDOW small
   ? CHR(7)
   @ 1,1 SAY "Order has ZERO phases,you will have to"
   @ 2,1 SAY "add some from the Edit Order Screen."
   @ 3,3 SAY "Press any key..."
   WAIT ""
   DEACTIVATE WINDOW small
   CLEAR
   RETURN
ENDIF (phases=0)
STORE phases TO phasevar
SELECT 1
* Setbrows/Endbrows not used here for obvious reasons!
SET MESSAGE TO "Enter Y for each Task Organization in this Order. Ctrl-W when
Done"
ACTIVATE WINDOW browser
BROWSE NOAPPEND NODELETE NOMENU FIELDS changestat, taskorgnam FREEZE changestat
COMPRESS
DEACTIVATE WINDOW browser
COPY TO ttemp FOR changestat
SELECT 18
COPY STRUCTURE TO temp
* Create one record for each Task Organization for each Phase of the Order
SELECT 11
USE temp
STORE 1 TO countvar
SET MESSAGE TO "Please wait, working..."
DO WHILE .T.
   APPEND FROM ttemp
   REPLACE ALL phase WITH countvar FOR phase=0
   countvar=countvar+1
   IF countvar>phasevar
      EXIT
```

```
      ENDIF (countvar>phasevar)
ENDDO (.T.)
REPLACE ALL NUMBER WITH orderlst->number
* Initialize Task Orgs for this order at 100 %
REPLACE ALL strength WITH 100
SELECT 11
USE
SELECT 18
APPEND FROM temp
SELECT 1
REPLACE ALL changestat WITH .F. FOR changestat
SET MESSAGE TO
KEYBOARD "{DNARROW}"
* CASE BAR()=4
*!**********************************************************************
*!
*!          Procedure: ENDORDR
*!
*!          Called by: ORDS_PRO             (procedure in CREATEOR.PRG)
*!                   : ORDT_PRO             (procedure in EDITORDE.PRG)
*!
*!**********************************************************************
PROCEDURE endordr
SELECT 18
SET ORDER TO profilenam
SEEK " "
IF FOUND()
   STORE "YES" TO yesnovar
   ACTIVATE WINDOW small
   ? CHR(7)
   @ 1,1 SAY "You have not assigned profiles to"
   @ 2,1 SAY "all units. Cancel exit? "
   @ ROW(),COL() GET yesnovar
   READ
   IF yesnovar="YES"
      DEACTIVATE WINDOW small
      RETURN
   ELSE
      DEACTIVATE WINDOW small
   ENDIF (yesnovar="YES")
ENDIF (FOUND())
CLOSE DATA
DEACTIVATE POPUP
RETURN
*!**********************************************************************
*!
*!          Procedure: STARTOR
*!
*!          Called by: CREATEOR.PRG
*!                   : EDITORDE.PRG
*!                   : ORDS_PRO             (procedure in CREATEOR.PRG)
*!                   : ORDT_PRO             (procedure in EDITORDE.PRG)
*!
*!              Calls: STARTHLP.PRG
*!
*!               Uses: CPROFILE.DBF
*!                   : TASKORG.DBF
*!                   : ORDERTAS.DBF
*!                   : ORDERLST.DBF
*!
```

```
*!        MDX files: CPROFILE.MDX
*!                 : TASKORG.MDX
*!                 : ORDERTAS.MDX
*!                 : ORDERLST.MDX
*!
*!*********************************************************************
PROCEDURE startor
STORE 1 TO statusvar
DO starthlp
SELECT 4
USE cprofile
SELECT 1
USE taskorg ORDER uniqueta
SELECT 18
USE ordertas
SELECT 3
USE orderlst
*!*********************************************************************
*!
*!        Procedure: DISPORD
*!
*!        Called by: CREATEO          (procedure in CREATEOR.PRG)
*!                 : GETORDER          (procedure in EDITORDE.PRG)
*!
*!*********************************************************************
PROCEDURE dispord
@ 17,2 TO 17,78
@ 19,2 SAY "Order Number:   "
@ ROW(),COL() SAY pnumberv
@ 21,2 SAY "  Order Name:   "
@ ROW(),COL() SAY namevar
*!*********************************************************************
*!
*!        Procedure: DISPORD2
*!
*!        Called by: VIEWORDE          (procedure in CREATEOR.PRG)
*!                 : ORDT_PRO          (procedure in EDITORDE.PRG)
*!                 : GETORDER          (procedure in EDITORDE.PRG)
*!
*!*********************************************************************
PROCEDURE dispord2
@ 1,2 SAY "Order Number:   "
@ ROW(),COL() SAY NUMBER
@ 2,2 SAY "  Order Name:   "
@ ROW(),COL() SAY name
@ 3,2 SAY "Description:   "
@ ROW(), COL() SAY descriptio
@ 4,2 SAY "Number of Phases:   "
@ ROW(), COL() SAY LTRIM(STR(phases))
********************************************************
* Popup for Mission Profile/Task Organization assignment choices
* BAR 1 OF Tomp_pop PROMPT "Assign one Profile to all Task Organizations and
Phases"
* BAR 2 OF Tomp_pop PROMPT "Assign one Profile to one Task Organization for all
Phases"
* BAR 3 OF Tomp_pop PROMPT "Assign one Profile to all Task Organizations for a
single Phase"
* BAR 4 OF Tomp_pop PROMPT "Assign Profiles to a Task Organization Phase by
Phase"
* BAR 5 OF Tomp_pop PROMPT "View Task Organizations and Profiles by Phase"
```

```
* BAR 6 OF Tomp_pop PROMPT "Return to Previous Menu"
* Procedures for tomp_pop
*!*****************************************************************
*!
*!          Procedure: TOMP_PRO
*!
*!          Called by: CREATEOR.PRG
*!                    : EDITORDE.PRG
*!
*!              Calls: 1PROALL              (procedure in CREATEOR.PRG)
*!                   : 1PROONE              (procedure in CREATEOR.PRG)
*!                   : 1PRO1PRI             (procedure in CREATEOR.PRG)
*!                   : PHASEBY              (procedure in CREATEOR.PRG)
*!                   : VIEWTOP              (procedure in CREATEOR.PRG)
*!
*!*****************************************************************
PROCEDURE tomp_pro
DO CASE
CASE BAR()=1
   DO 1proall
CASE BAR()=2
   DO 1proone
CASE BAR()=3
   DO 1pro1pri
CASE BAR()=4
   DO phaseby
CASE BAR()=5
   DO viewtop
CASE BAR()=6
   DEACTIVATE POPUP
ENDCASE
*CASE BAR()=1
*!*****************************************************************
*!
*!          Procedure: 1PROALL
*!
*!          Called by: TOMP_PRO              (procedure in CREATEOR.PRG)
*!
*!              Calls: FINDPRO               (procedure in CREATEOR.PRG)
*!
*!*****************************************************************
PROCEDURE 1proall
SET MESSAGE TO "Ctrl-W to Choose Profile for all Task Organizations and Phases"
DO findpro
SELECT 18
REPLACE ALL profilenam WITH cprofile->profilenam FOR ordertas->number=pnumberv
CLEAR
*CASE BAR()=2
*!*****************************************************************
*!
*!          Procedure: 1PROONE
*!
*!          Called by: TOMP_PRO              (procedure in CREATEOR.PRG)
*!
*!              Calls: SETBROWS.PRG
*!                   : ENDBROWS.PRG
*!                   : FINDPRO               (procedure in CREATEOR.PRG)
*!
*!*****************************************************************
PROCEDURE 1proone
```

92

```
* Choose Task Organization
SELECT 18
SET ORDER TO NUMBER
SET FILTER TO NUMBER=pnumberv
GO TOP
DO setbrows
BROWSE NOAPPEND NODELETE NOEDIT NOMENU FIELDS taskorgnam COMPRESS
DO endbrows
STORE taskorgnam TO namevar
* DO findta
* Choose profile
SET MESSAGE TO "Ctrl-W to Choose Profile for this Task Organizations for all
Phases"
SET FILTER TO
SET ORDER TO taskorgnam
DO findpro
CLEAR
SELECT 18
REPLACE ALL profilenam WITH cprofile->profilenam FOR ordertas->number=pnumberv;
    .AND. taskorgnam=namevar
SET MESSAGE TO
CLEAR
*CASE BAR()=3
*!*************************************************************************
*!
*!        Procedure: 1PRO1PRI
*!
*!        Called by: TOMP_PRO              (procedure in CREATEOR.PRG)
*!
*!           Calls: FINDPRO                (procedure in CREATEOR.PRG)
*!
*!*************************************************************************
PROCEDURE 1pro1pri
SELECT 3
STORE 0 TO numbe2var
STORE phases TO numberva
SET MESSAGE TO "Ctrl-W to Choose Profile for all Task Organizations and Phases"
DO findpro
ACTIVATE WINDOW small
@ 1,1 SAY "Enter Phase Number to use selected"
@ 2,1 SAY "Profile for: "
@ ROW(),COL() GET numbe2var VALID numbe2var<=numberva .AND. numbe2var>=0;
    ERROR "Must be between 0 and "+LTRIM(STR(numberva))+". 0 cancels."
@ 3,1 SAY "There are "+LTRIM(STR(numbervar))+" Phases in this Order"
READ
IF numbe2var=0
    CLEAR
    ? CHR(7)
    @ 1,2 SAY "No Phase selected, action cancelled!"
    @ 3,3 SAY "Press any key..."
    WAIT ""
    DEACTIVATE WINDOW small
    CLEAR
    RETURN
ENDIF (numbe2var=0)
DEACTIVATE WINDOW small
SELECT 18
REPLACE ALL profilenam WITH cprofile->profilenam;
    FOR ordertas->number=pnumberv .AND. phase=numbe2var
CLEAR
```

93

```
*CASE BAR()=4
*!********************************************************************
*!
*!        Procedure: PHASEBY
*!
*!        Called by: TOMP_PRO              (procedure in CREATEOR.PRG)
*!
*!            Calls: VIEWTOP               (procedure in CREATEOR.PRG)
*!                 : FINDPRO               (procedure in CREATEOR.PRG)
*!
*!********************************************************************
PROCEDURE phaseby
DO viewtop
DO findpro
SELECT 18
REPLACE profilenam WITH cprofile->profilenam
*DO disptas
*DO dispprof
SET FILTER TO
*CASE BAR()=5
*!********************************************************************
*!
*!        Procedure: VIEWTOP
*!
*!        Called by: TOMP_PRO              (procedure in CREATEOR.PRG)
*!                 : PHASEBY               (procedure in CREATEOR.PRG)
*!
*!            Calls: SETBROWS.PRG
*!                 : ENDBROWS.PRG
*!
*!********************************************************************
PROCEDURE viewtop
SELECT 18
* Put in taskorgnam+str(phase) order
SET ORDER TO taskphas
SET FILTER TO NUMBER=pnumberv
GO TOP
ACTIVATE WINDOW browser
DO setbrows
BROWSE NOEDIT NOAPPEND NODELETE NOMENU FIELDS phase, profilenam, taskorgnam
DO endbrows
SET FILTER TO
SET ORDER TO taskorgnam
DEACTIVATE WINDOW browser
*!********************************************************************
*!
*!        Procedure: DISPPROF
*!
*!        Called by: NEWPROFI             (procedure in CREATEPR.PRG)
*!                 : VIEWPROF             (procedure in CREATEPR.PRG)
*!                 : SELEPROF             (procedure in EDITPROF.PRG)
*!                 : CHANGEAP             (procedure in EDITPROF.PRG)
*!                 : CHANGEAC             (procedure in EDITPROF.PRG)
*!                 : CHANGEI              (procedure in EDITPROF.PRG)
*!                 : CHANGEIF             (procedure in EDITPROF.PRG)
*!                 : CHANGECI             (procedure in EDITPROF.PRG)
*!                 : CHANGEH2             (procedure in EDITPROF.PRG)
*!                 : CHANGEME             (procedure in EDITPROF.PRG)
*!                 : SELECTPRO            (procedure in DELPROF.PRG)
*!                 : FINDPRO              (procedure in CREATEOR.PRG)
```

```
*!
*!*******************************************************************
PROCEDURE dispprof
SELECT 4
@12,1 SAY "Profile Name:"
@12,17 SAY profilenam PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
@13,1 SAY "Description: "
@13,17 SAY profiledes PICTURE
"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
@14,1 SAY "Area"
@14,6 SAY "Profile:"
@14,17 SAY profile PICTURE "XXXXXXXXXXXXXXX"
@15, 1 SAY "Intensity:"
@15,17 SAY INTENSITY PICTURE "XXXXXXXX"
@15,27 SAY "Factor"
@15,35 SAY factor PICTURE "999.99"
@16,1 SAY "Ammo Consumption Posture: "
@ROW(),COL() SAY posture
*!*******************************************************************
*!
*!
*!          Procedure: FINDPRO
*!
*!          Called by: 1PROALL              (procedure in CREATEOR.PRG)
*!                   : 1PROONE              (procedure in CREATEOR.PRG)
*!                   : 1PRO1PRI             (procedure in CREATEOR.PRG)
*!                   : PHASEBY              (procedure in CREATEOR.PRG)
*!
*!             Calls: SETBROWS.PRG
*!                  : ENDBROWS.PRG
*!                  : DISPPROF             (procedure in CREATEPR.PRG)
*!
*!*******************************************************************
PROCEDURE findpro
SELECT 4
ACTIVATE WINDOW browser
GO TOP
DO setbrows
BROWSE NOMENU NOEDIT NOAPPEND NODELETE ;
    FIELDS profilenam,profiledes FREEZE profilenam COMPRESS
DO endbrows
DEACTIVATE WINDOW browser
SET CURSOR OFF
CLEAR
DO WHILE LASTKEY()<>23
    DO dispprof
    IF RECCOUNT()=1
        @ 23,30 SAY "Crtl-W when Done"
    ELSE
        @ 23,15 SAY "Page-Up or Page-Down to View, Ctrl-W when Done"
    ENDIF (RECCOUNT()=1)
    IF LASTKEY()=3
        SKIP
        IF EOF()
            GO TOP
        ENDIF (EOF())
    ENDIF (LASTKEY()=3)
    IF LASTKEY()=18
        SKIP -1
        IF BOF()
            GO BOTTOM
```

```
        ENDIF (BOF())
     ENDIF (LASTKEY()=18)
     WAIT ""
 ENDDO (LASTKEY()<>23)
 SET CURSOR ON
 SET MESSAGE TO
 *!*************************************************************************
 *!
 *!        Procedure: DISPTAS
 *!
 *!*************************************************************************
 PROCEDURE disptas
 @ 17,2 TO 17,78
 @ 21,2 SAY "Task Organization:   "
 @ ROW(),COL() SAY namevar
 *!*************************************************************************
 *!
 *!        Procedure: FINDTA
 *!
 *!        Called by: EDITTASK.PRG
 *!                 : COPT_PRO          (procedure in CREATETC.PRG)
 *!                 : EDIT_PRO          (procedure in EDITTASK.PRG)
 *!                 : DELT_PRO          (procedure in DELTASK.PRG)
 *!
 *!           Calls: SETBROWS.PRG
 *!                 : ENDBROWS.PRG
 *!                 : DISPTASK          (procedure in CREATETA.PRG)
 *!
 *!*************************************************************************
 PROCEDURE findta
 SELECT 3
 STORE NUMBER TO pnumberv
 SELECT 18
 *STORE RECNO() TO recnovar
 SET ORDER TO uniqueta
 SET FILTER TO NUMBER=pnumberv
 GO TOP
 ACTIVATE WINDOW browser
 DO setbrows
 BROWSE FIELDS taskorgnam NOMENU NOEDIT NOAPPEND NODELETE COMPRESS
 DO endbrows
 DEACTIVATE WINDOW browser
 STORE taskorgnam TO namevar
 SET FILTER TO
 SET ORDER TO NUMBER
 *GO recnovar
 *DO disptas

 *: EOF: CREATEOR.PRG
```

```
*:********************************************************************
*:                                                    '
*: Procedure file: D:\SURE\CREATEPR.PRG
*:
*:   Procs & Fncts: STARTPRO
*:                 : CREP_PRO
*:                 : NEWPROFI
*:                 : VIEWPROF
*:                 : ENDCREP
*:                 : DISPPROF
*:
*:         Set by: PROF_PRO          (procedure in SURE.PRG)
*:
*:         Calls: STARTPRO           (procedure in CREATEPR.PRG)
*:              : CREP_PRO           (procedure in CREATEPR.PRG)
*:
*:         Documented 04/16/94 at 11:01             SNAP!  version 5.00
*:********************************************************************
* createpro.prg
CLEAR
STORE PROGRAM() TO progvar
SET PROCEDURE TO instprof
DO startpro
DEFINE POPUP crep_pop FROM 1,14
DEFINE BAR 1 OF crep_pop PROMPT "Create New Mission Parameter Set"
DEFINE BAR 2 OF crep_pop PROMPT "View Mission Parameter Sets"
DEFINE BAR 3 OF crep_pop PROMPT "Return to Main Menu"
ON SELECTION POPUP crep_pop DO crep_pro
ON BAR 1 OF crep_pop DO crep_bar1
ON BAR 2 OF crep_pop DO crep_bar2
ON BAR 3 OF crep_pop DO crep_bar3
DO WHILE PROMPT()<>"Return to Main Menu"
   ACTIVATE POPUP crep_pop
ENDDO (PROMPT()<>"Return to Main Menu")
*!********************************************************************
*!
*!      Procedure: CREP_PRO
*!
*!      Called by: CREATEPR.PRG
*!
*!         Calls: NEWPROFI           (procedure in CREATEPR.PRG)
*!              : VIEWPROF           (procedure in CREATEPR.PRG)
*!              : ENDCREP            (procedure in CREATEPR.PRG)
*!
*!********************************************************************
PROCEDURE crep_pro
DO CASE
CASE BAR()=1
   DO newprofi
CASE BAR()=2
   DO viewprof
CASE BAR()=3
   DO endcrep
ENDCASE
* Procedures for CreP_pop
* CASE BAR() = 1
*!********************************************************************
*!
*!      Procedure: NEWPROFI
*!
```

97

```
*!         Called by: CREP_PRO         (procedure in CREATEPR.PRG)
*!
*!            Calls: SETBROWS.PRG
*!                 : ENDBROWS.PRG
*!                 : DISPPROF          (procedure in CREATEPR.PRG)
*!
*!             Uses: TEMP.DBF
*!
*!*****************************************************************************
PROCEDURE newprofi
SELECT 4
COPY STRUCTURE TO temp
SELECT 11
USE temp
APPEND BLANK
*SET FORMAT TO addprofi
KEYBOARD CHR(22)
SET MESSAGE TO
CLEAR
DO WHILE LASTKEY()<>23
   @11,60 SAY "Parameter Set Names"
   @12,1 SAY "    Set Name:"
   @12,17 GET profilenam PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
;
       MESSAGE "This cannot be modified later!"
   @12,60 SAY "are permanent!"
   @13,1 SAY "Description: "
   @13,17 GET profiledes PICTURE
"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
   @14,1 SAY "Area"
   @14,6 SAY "Profile:"
   @14,17 SAY profile PICTURE "XXXXXXXXXXXXXXX"
   @15, 1 SAY "Intensity:"
   @15,17 SAY INTENSITY PICTURE "XXXXXXXX"
   @15,27 SAY "Factor:"
   @15,35 SAY factor PICTURE "999.99"
   @16,1 SAY "Ammo Consumption Posture: "
   @ROW(),COL() SAY posture
   @ 21,19 SAY "Enter Name and Description, then Ctrl-W"
   READ
ENDDO (LASTKEY()<>23)
* Make sure a name was added
IF profilenam=" "
   ACTIVATE WINDOW small
   ? CHR(7)
   @ 1,2 SAY "Set NOT Named  "
   @ 2,2 SAY "Set NOT added!    "
   @ 3,3 SAY "Press any key..."
   WAIT ""
   SELECT 11
   ZAP
   SELECT 11
   USE
   DEACTIVATE WINDOW small
   CLEAR
   RETURN
ENDIF (profilenam=" ")
* Make sure name is not a duplicate
SELECT 4
SEEK temp->profilenam
```

```
IF .NOT. EOF()
    ACTIVATE WINDOW small
    ? CHR(7)
    @ 1,2 SAY "Duplicate Set Name"
    @ 2,2 SAY "Set NOT added!"
    @ 3,3 SAY "Press any key..."
    WAIT ""
    SELECT 11
    ZAP
    SELECT 11
    USE
    DEACTIVATE WINDOW small
    CLEAR
    RETURN
ENDIF (.NOT. EOF())
ACTIVATE WINDOW editor
* Get CASCOM Area Profile
SELECT 7
DO setbrows
SET MESSAGE TO "Choose an Area Profile, then Enter to Select"
GO TOP
BROWSE NOMENU NOAPPEND NODELETE NOEDIT FIELDS profile
SELECT 11
REPLACE temp->profile WITH profiles->profile
ACTIVATE SCREEN
@16,17 SAY profile PICTURE "XXXXXXXXXXXXXXX"
ACTIVATE WINDOW editor
* Get Combat Posture
SET MESSAGE TO "Choose a Combat Posture for Ammunition Consumption, then Enter
to Select"
SELECT 17
GO TOP
BROWSE NOMENU NOAPPEND NODELETE NOEDIT
REPLACE temp->posture WITH posture->posture
* Get Intensity
SET MESSAGE TO "Choose an Intensity for this Parameter Set, then Enter to
Select"
SELECT 10
GO TOP
BROWSE NOMENU NOAPPEND NODELETE NOEDIT FIELDS INTENSITY
SELECT 11
REPLACE temp->intensity WITH intensty->intensity
REPLACE temp->factor WITH intensty->factor
SET MESSAGE TO
* Class I
SELECT 24
SET MESSAGE TO "Choose Type of Ration for Breakfast Meal, then Enter to Select"
GO TOP
BROWSE NOMENU NOAPPEND NODELETE NOEDIT FIELDS ration, unit, plan_fact
REPLACE temp->cl_i_1 WITH class_i->ration
SELECT 24
SET MESSAGE TO "Choose Type of Ration for Lunch Meal, then Enter to Select"
GO TOP
BROWSE NOMENU NOAPPEND NODELETE NOEDIT FIELDS ration, unit, plan_fact
REPLACE temp->cl_i_2 WITH class_i->ration
SELECT 24
SET MESSAGE TO "Choose Type of Ration for Dinner Meal, then Enter to Select"
GO TOP
BROWSE NOMENU NOAPPEND NODELETE NOEDIT FIELDS ration, unit, plan_fact
REPLACE temp->cl_i_3 WITH class_i->ration
```

99

```
* Select Water Parameters
SELECT 23
SET MESSAGE TO "Choose Water Use Parameters, then Enter to Select"
BROWSE NOMENU NOAPPEND NODELETE NOEDIT FIELDS gal_man_da, echelon,climate
REPLACE temp->water_name WITH water->water_name
* Select Medical Class VIII Parameters
SELECT 22
SET MESSAGE TO "Choose Medical Class VIII Consumption Parameters, then Enter to
Select"
BROWSE NOMENU NOAPPEND NODELETE NOEDIT FIELDS plan_fact,echelon,INTENSITY
DO endbrows
REPLACE temp->med_name WITH medical->med_name
SELECT 11
USE
SELECT 4
APPEND FROM temp
DEACTIVATE WINDOW editor
DO dispprof
SELECT 11
USE temp
ZAP
SELECT 11
USE
SELECT 4
* CASE BAR() = 2
*!****************************************************************
*!
*!       Procedure: VIEWPROF
*!
*!       Called by: CREP_PRO              (procedure in CREATEPR.PRG)
*!
*!           Calls: SETBROWS.PRG
*!                : ENDBROWS.PRG
*!                : DISPPROF               (procedure in CREATEPR.PRG)
*!
*!****************************************************************
PROCEDURE viewprof
SELECT 4
* Make sure there are some records to view
IF RECCOUNT()=0
   ACTIVATE WINDOW small
   ? CHR(7)
   @ 1,2 SAY "You must have at"
   @ 2,2 SAY "least one Set!"
   @ 3,3 SAY "Press any key..."
   WAIT ""
   DEACTIVATE WINDOW small
   CLEAR
   RETURN
ENDIF (RECCOUNT()=0)
ACTIVATE WINDOW browser
GO TOP
*        SET MESSAGE TO "Choose Set to View"
DO setbrows
BROWSE NOMENU NOEDIT NOAPPEND NODELETE FIELDS profilenam,profiledes FREEZE
profilenam COMPRESS
DO endbrows
DEACTIVATE WINDOW browser
SET CURSOR OFF
DO WHILE LASTKEY()<>23
```

100

```
        DO dispprof
        IF RECCOUNT()=1
           @ 23,30 SAY "Crtl-W when Done"
        ELSE
           @ 23,15 SAY "Page-Up or Page-Down to View, Ctrl-W when Done"
        ENDIF (RECCOUNT()=1)
        IF LASTKEY()=3
           SKIP
           IF EOF()
              GO TOP
           ENDIF (EOF())
        ENDIF (LASTKEY()=3)
        IF LASTKEY()=18
           SKIP -1
           IF BOF()
              GO BOTTOM
           ENDIF (BOF())
        ENDIF (LASTKEY()=18)
        WAIT ""
ENDDO (LASTKEY()<>23)
SET CURSOR ON
SET MESSAGE TO
CLEAR
DO dispprof
* CASE BAR() = 3
*!********************************************************************
*!
*!        Procedure: ENDCREP
*!
*!        Called by: CREP_PRO            (procedure in CREATEPR.PRG)
*!                 : EDIP_PRO            (procedure in EDITPROF.PRG)
*!
*!********************************************************************
PROCEDURE endcrep
CLOSE DATA
DEACTIVATE POPUP
RETURN
*!********************************************************************
*!
*!        Procedure: STARTPRO
*!
*!        Called by: CREATEPR.PRG
*!                 : CRECUSPR.PRG
*!
*!            Calls: STARTHLP.PRG
*!
*!             Uses: MEDICAL.DBF
*!                 : WATER.DBF
*!                 : CLASS_I.DBF
*!                 : POSTURE.DBF
*!                 : INTENSTY.DBF
*!                 : PROFILES.DBF
*!                 : CPROFILE.DBF
*!
*!        MDX files: MEDICAL.MDX
*!                 : WATER.MDX
*!                 : CLASS_I.MDX
*!                 : INTENSTY.MDX
*!                 : PROFILES.MDX
*!                 : CPROFILE.MDX
```

```
*!
*!*************************************************************
PROCEDURE startpro
DO starthlp
SELECT 22
USE medical
SELECT 23
USE water
SELECT 24
USE class_i
SELECT 17
USE posture
SELECT 10
USE intensty
SELECT 7
USE profiles
SELECT 4
USE cprofile
SET ORDER TO profilenam
STORE SPACE(60) TO namevar
*!*************************************************************
*!
*!        Procedure: DISPPROF
*!
*!        Called by: NEWPROFI          (procedure in CREATEPR.PRG)
*!                 : VIEWPROF          (procedure in CREATEPR.PRG)
*!                 : SELEPROF          (procedure in EDITPROF.PRG)
*!                 : CHANGEAP          (procedure in EDITPROF.PRG)
*!                 : CHANGEAC          (procedure in EDITPROF.PRG)
*!                 : CHANGEI           (procedure in EDITPROF.PRG)
*!                 : CHANGEIF          (procedure in EDITPROF.PRG)
*!                 : CHANGECI          (procedure in EDITPROF.PRG)
*!                 : CHANGEH2          (procedure in EDITPROF.PRG)
*!                 : CHANGEME          (procedure in EDITPROF.PRG)
*!                 : SELECTPRO         (procedure in DELPROF.PRG)
*!                 : FINDPRO           (procedure in CREATEOR.PRG)
*!*************************************************************
PROCEDURE dispprof
@12,1 SAY "          Name:"
@12,17 SAY profilenam PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
@13,1 SAY "Description: "
@13,17 SAY profiledes PICTURE
"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
@14,1 SAY "Area"
@14,6 SAY "Profile:"
@14,17 SAY profile PICTURE "XXXXXXXXXXXXXXX"
@15, 1 SAY "Intensity:"
@15,17 SAY INTENSITY PICTURE "XXXXXXXX"
@15,27 SAY "Factor"
@15,35 SAY factor PICTURE "999.99"
@16,1 SAY "Ammo Consumption Posture: "
@ROW(),COL() SAY posture
@17,1 SAY "Class I Ration Cycle    "
@ ROW(), COL() SAY cl_i_1+" "+cl_i_2+" "+cl_i_3
@18,1 SAY "Water Use Parameters:   "
@ ROW(), COL() SAY water_name
@19,1 SAY "Class VIII Parameters: "
@ ROW(), COL() SAY med_name

*: EOF: CREATEPR.PRG
```

```
*:******************************************************************
*:
*:  Procedure file: D:\SURE\CREATESR.PRG
*:
*:   Procs & Fncts: STARTCSR
*:                : CSRC_PRO
*:                : SEEKSRC
*:                : COPYSRC
*:                : COPYCSRC
*:                : VIEWUNIT
*:                : CSRCEND
*:                : CSRCENDA
*:                : COPYLINS
*:                : DISPUNIT
*:                : ADDDATA
*:
*:          Set by: SRC_PRO              (procedure in SURE.PRG)
*:
*:           Calls: STARTCSR             (procedure in CREATESR.PRG)
*:                : CSRC_PRO             (procedure in CREATESR.PRG)
*:
*:          Documented 04/16/94 at 11:01            SNAP!  version 5.00
*:******************************************************************
* createsr.prg
* creates user SRC using CASCOM supplied or previous user SRC as basis.
* PRIVATE Variables
STORE PROGRAM() TO progvar
STORE "NO " TO yesnovar
STORE SPACE(9) TO srcvar
SET PROCEDURE TO instunit
DO startcsr
DEFINE POPUP csrc_pop FROM 1,0
DEFINE BAR 1 OF csrc_pop PROMPT "Select SRC (Browse)"
DEFINE BAR 2 OF csrc_pop PROMPT "Select SRC (Find SRC with SRC Number)"
DEFINE BAR 3 OF csrc_pop PROMPT "Add New Unit from SRC or Copied Unit"
DEFINE BAR 4 OF csrc_pop PROMPT "Copy Unit from Existing Unit"
DEFINE BAR 5 OF csrc_pop PROMPT "View Existing Units"
DEFINE BAR 6 OF csrc_pop PROMPT "Return to Main Menu"
ON SELECTION POPUP csrc_pop DO csrc_pro
ON BAR 1 OF csrc_pop DO csrc_bar1
ON BAR 2 OF csrc_pop DO csrc_bar2
ON BAR 3 OF csrc_pop DO csrc_bar3
ON BAR 4 OF csrc_pop DO csrc_bar4
ON BAR 5 OF csrc_pop DO csrc_bar5
ON BAR 6 OF csrc_pop DO csrc_bar6
DO WHILE PROMPT()<>"Return to Main Menu"
   ACTIVATE POPUP csrc_pop
ENDDO (PROMPT()<>"Return to Main Menu")
SELECT 2
*!******************************************************************
*!
*!        Procedure: CSRC_PRO
*!
*!        Called by: CREATESR.PRG
*!
*!            Calls: COPYSRC              (procedure in CREATESR.PRG)
*!                : SEEKSRC              (procedure in CREATESR.PRG)
*!                : CSRCENDA             (procedure in CREATESR.PRG)
*!                : COPYCSRC             (procedure in CREATESR.PRG)
*!                : VIEWUNIT             (procedure in CREATESR.PRG)
```
103

```
*!                       : CSRCEND          (procedure in CREATESR.PRG)
*!
*!*******************************************************************
PROCEDURE csrc_pro
DO CASE
CASE BAR() = 1
    CLEAR
    DO copysrc
    @ 3,39 TO 8,78
    @ 4,40 SAY "<-CHOOSE THIS OPTION TO CREATE A UNIT!"
    @ 5,40 SAY "You have selected an SRC, but must"
    @ 6,40 SAY "create the Unit from it to copy the"
    @ 7,40 SAY "equipment LINs and quantities."
CASE BAR() = 2
    DO seeksrc
    * DONE Save SRC and add another
CASE BAR() = 3
    DO csrcenda
CASE BAR() = 4
    IF RECCOUNT()>0
        DO copycsrc
        @ 3,39 TO 8,78
        @ 4,40 SAY "<-CHOOSE THIS OPTION TO CREATE A UNIT!"
        @ 5,40 SAY "You have selected an SRC, but must"
        @ 6,40 SAY "create the Unit from it to copy the"
        @ 7,40 SAY "equipment LINs and quantities."
    ENDIF (RECCOUNT()>0)
    * View Existing Units
CASE BAR() = 5
    DO viewunit
    * DONE Save SRC and end
CASE BAR() = 6
    DO csrcend
ENDCASE
* Seek SRC
*!*******************************************************************
*!
*!        Procedure: SEEKSRC
*!
*!        Called by: CSRC_PRO          (procedure in CREATESR.PRG)
*!
*!            Calls: COPYSRC           (procedure in CREATESR.PRG)
*!
*!*******************************************************************
PROCEDURE seeksrc
ACTIVATE WINDOW small
SET CONFIRM ON
@ 1,1 SAY "Enter SRC to find: "
@ ROW(), COL() GET srcvar
READ
SET CONFIRM OFF
DEACTIVATE WINDOW small
SELECT 5
SET ORDER TO src
SET NEAR ON
SEEK srcvar
SET NEAR OFF
IF EOF()
    ACTIVATE WINDOW small
    @ 1,1 SAY "Not found.. Press any key."
```

```
     WAIT ""
     DEACTIVATE WINDOW small
ELSE
     DO copysrc
ENDIF (EOF())
* Copy new SRC from CASCOM SRC
*!***********************************************************************
*!
*!        Procedure: COPYSRC
*!
*!        Called by: CSRC_PRO              (procedure in CREATESR.PRG)
*!                 : SEEKSRC               (procedure in CREATESR.PRG)
*!
*!           Calls: SETBROWS.PRG
*!                 : ENDBROWS.PRG
*!                 : DISPUNIT              (procedure in CREATESR.PRG)
*!
*!***********************************************************************
PROCEDURE copysrc
CLEAR
SELECT 5
*ACTIVATE WINDOW browser
DO setbrows
BROWSE NOEDIT NOAPPEND NOORGANIZE NODELETE
DO endbrows
SELECT 11
REPLACE temp->src WITH srclist->src
REPLACE temp->unit_desig WITH srclist->unit_desig
REPLACE temp->strength WITH srclist->strength
REPLACE temp->unitname WITH srclist->unit_desig
CLEAR
DO WHILE .T.
     @ 1,4 SAY "Original"
     @ 1,13 SAY "SRC DATA:"
     @ 1,23 SAY unit_desig PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
     @ 1,65 SAY src PICTURE "XXXXXXXXX"
     @ 2,2 SAY
"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"
     @ 4,1 SAY "Unit Name: "
     @ 4,12 GET unitname PICTURE
"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
     @ 6,28 SAY "Authorized Strength: "
     @ 6,49 GET strength PICTURE "999999"
     @ 8,12 SAY "Organizational relationships and logistical caluculations"
     @ 9,12 SAY "are based on the Unit Name field.  You will NOT be allowed"
     @ 10,12 SAY "to change the Unit Name once you leave this screen, although"
     @ 11,12 SAY "you will be able to copy from a previously defined SRC."
     @ 17,31 SAY "CTRL-W when Done!"
     READ
     * Error check to make sure unitname is not a duplicate
     STORE temp->unitname TO fieldvar
     SELECT 2
     IF KEYMATCH(fieldvar,TAGNO("UNITNAME"))                   &&FOUND()
         ? CHR(7)
         SET MESSAGE TO "You have duplicated a Unit Name already in use!"
         SELECT 11
     ELSE
         EXIT
     ENDIF (KEYMATCH(fieldvar,TAGNO("UNITNAME"))               &&FOUND())
ENDDO (.T.)
```

105

```
SET MESSAGE TO
CLEAR
DO dispunit
* Copy new SRC from previous custom SRC
*!*******************************************************************
*!
*!        Procedure: COPYCSRC
*!
*!        Called by: CSRC_PRO                 (procedure in CREATESR.PRG)
*!
*!            Calls: SETBROWS.PRG
*!                 : ENDBROWS.PRG
*!                 : DISPUNIT                  (procedure in CREATESR.PRG)
*!
*!*******************************************************************
PROCEDURE copycsrc
CLEAR
SELECT 2
ACTIVATE WINDOW browser
DO setbrows
BROWSE NOEDIT NOAPPEND NOORGANIZE NODELETE
DO endbrows
DEACTIVATE WINDOW browser
SELECT 11
REPLACE temp->src WITH csrclist->src
REPLACE temp->unit_desig WITH csrclist->unit_desig
REPLACE temp->strength WITH csrclist->strength
REPLACE temp->unitname WITH csrclist->unit_desig
CLEAR
DO WHILE .T.
   SELECT 11
   @ 1,4 SAY "Original"
   @ 1,13 SAY "SRC DATA:"
   @ 1,23 SAY unit_desig PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
   @ 1,65 SAY src PICTURE "XXXXXXXXX"
   @ 2,2 SAY
"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"
   @ 4,1 SAY "Unit Name: "
   @ 4,12 GET unitname PICTURE
"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
   @ 6,28 SAY "Authorized Strength: "
   @ 6,49 GET strength PICTURE "999999"
   @ 8,12 SAY "Organizational relationships and logistical calculations"
   @ 9,12 SAY "are based on the Unit Name field.  You will NOT be allowed"
   @ 10,12 SAY "to change the Unit Name once you leave this screen, although"
   @ 11,12 SAY "you will be able to copy from a previously defined SRC."
   @ 17,31 SAY "CTRL-W when Done!"
   READ
   * Error check to make sure unitname is not a duplicate
   STORE temp->unitname TO fieldvar
   SELECT 2
   IF KEYMATCH(fieldvar,TAGNO("UNITNAME"))
      ? CHR(7)
      SET MESSAGE TO "You have duplicated a Unit Name already in use!"
      SELECT 11
   ELSE
      EXIT
   ENDIF (KEYMATCH(fieldvar,TAGNO("UNITNAME")))
ENDDO (.T.)
SET MESSAGE TO
```

106

```
CLEAR
DO dispunit
*!*******************************************************************
*!
*!        Procedure: VIEWUNIT
*!
*!        Called by: CSRC_PRO            (procedure in CREATESR.PRG)
*!
*!            Calls: SETBROWS.PRG
*!                 : ENDBROWS.PRG
*!
*!*******************************************************************
PROCEDURE viewunit
SELECT 2
IF RECCOUNT()=0
   ACTIVATE WINDOW small
   ? CHR(7)
   @ 2,1 SAY "Select SRC or Unit first!"
   @ 3,1 SAY "Press any key..."
   WAIT ""
   DEACTIVATE WINDOW small
   RETURN
ENDIF (RECCOUNT()=0)
GO TOP
DO setbrows
*    SET MESSAGE TO "End for 1st Record, Home for Last Record"
ACTIVATE WINDOW browser
BROWSE NOEDIT NOMENU NOAPPEND NODELETE FIELDS src,unitname
DEACTIVATE WINDOW browser
DO endbrows
SELECT 11
* DONE
*!*******************************************************************
*!
*!        Procedure: CSRCEND
*!
*!        Called by: CSRC_PRO            (procedure in CREATESR.PRG)
*!
*!            Calls: ADDDATA             (procedure in CREATESR.PRG)
*!                 : COPYLINS            (procedure in CREATESR.PRG)
*!
*!*******************************************************************
PROCEDURE csrcend
IF temp->src<>" "
   ACTIVATE WINDOW small
   @ 1,1 SAY "Do you want to create a unit"
   @ 2,1 SAY "From the data you selected? "
   @ 3,1 SAY "Enter YES to do so! "
   @ ROW(), COL() GET yesnovar PICTURE "!!!"
   READ
   DEACTIVATE WINDOW small
   IF yesnovar="YES"
      DO adddata
      DO copylins
   ELSE
      SELECT temp
      ZAP
   ENDIF (yesnovar="YES")
ENDIF (temp->src<>" ")
CLOSE DATA
```

107

```
SET FORMAT TO
DEACTIVATE POPUP
RETURN
* DONE and add another
*!*****************************************************************
*!
*!        Procedure: CSRCENDA
*!
*!        Called by: CSRC_PRO            (procedure in CREATESR.PRG)
*!
*!            Calls: ADDDATA             (procedure in CREATESR.PRG)
*!                 : COPYLINS            (procedure in CREATESR.PRG)
*!                 : STARTCSR            (procedure in CREATESR.PRG)
*!
*!*****************************************************************
PROCEDURE csrcenda
IF temp->src=" "
   ACTIVATE WINDOW small
   ? CHR(7)
   @ 2,1 SAY "You must create a unit first!"
   @ 3,1 SAY "Press any key..."
   WAIT ""
   DEACTIVATE WINDOW small
   RETURN
ELSE
   DO adddata
   DO copylins
   DO startcsr
ENDIF (temp->src=" ")
KEYBOARD "{UPARROW}"
*KEYBOARD "{UPARROW}"
*KEYBOARD "{UPARROW}"
CLEAR
*!*****************************************************************
*!
*!        Procedure: STARTCSR
*!
*!        Called by: CREATESR.PRG
*!                 : CSRCENDA            (procedure in CREATESR.PRG)
*!
*!            Calls: STARTHLP.PRG
*!
*!             Uses: CSRCLIST.DBF
*!                 : TEMP.DBF
*!                 : SRCLIST.DBF
*!                 : SRCLINQT.DBF
*!                 : CSRCLINQ.DBF
*!
*!        MDX files: CSRCLIST.MDX
*!                 : SRCLIST.MDX
*!                 : SRCLINQT.MDX
*!                 : CSRCLINQ.MDX
*!
*!*****************************************************************
PROCEDURE startcsr
CLEAR
CLOSE DATA
DO starthlp
SELECT 2
USE csrclist
```

```
COPY STRUCTURE TO temp
SELECT 5
USE srclist
SELECT 6
USE srclinqt
SELECT 14
USE csrclinq
SELECT 11
USE temp
ZAP
APPEND BLANK
* Add LINs to csrclinq.dbf for this CSRC
*!*****************************************************************
*!
*!      Procedure: COPYLINS
*!
*!      Called by: CSRCEND              (procedure in CREATESR.PRG)
*!               : CSRCENDA             (procedure in CREATESR.PRG)
*!
*!           Uses: TTEMP.DBF
*!               : TTTEMP.DBF
*!
*!*****************************************************************
PROCEDURE copylins
@ 15,35 SAY "Please wait, copying LIN information..."
SELECT 14
COPY STRUCTURE TO ttemp
SELECT 6
COPY TO tttemp FOR srclinqt->src=temp->src
SELECT 15
USE ttemp
APPEND FROM tttemp
REPLACE ALL ttemp->unitname WITH temp->unitname
SELECT 15
USE
SELECT 14
APPEND FROM ttemp
@ 5,35 SAY "                            "
SET MESSAGE TO
* Display Unit data on screen
*!*****************************************************************
*!
*!      Procedure: DISPUNIT
*!
*!      Called by: COPYSRC              (procedure in CREATESR.PRG)
*!               : COPYCSRC             (procedure in CREATESR.PRG)
*!               : ADDDATA              (procedure in CREATESR.PRG)
*!               : SELECTUNI            (procedure in EDITUNIT.PRG)
*!               : SELECTUN2            (procedure in EDITUNIT.PRG)
*!               : CHANGELI             (procedure in EDITUNIT.PRG)
*!               : ADDLINCA             (procedure in EDITUNIT.PRG)
*!               : DELLIN               (procedure in EDITUNIT.PRG)
*!               : CHANGEEH             (procedure in EDITUNIT.PRG)
*!               : EDITSTR              (procedure in EDITUNIT.PRG)
*!               : UNITDATA             (procedure in CUSTUNIT.PRG)
*!
*!*****************************************************************
PROCEDURE dispunit
@ 16,1 TO 16,78
@ 18,5 SAY "SRC:      "+temp->src
```

```
@ 20,5 SAY "Unit:    "+temp->unitname
* Add SRC data to CSRClist and Cunit dbf's
*!*************************************************************
*!
*!        Procedure: ADDDATA
*!
*!        Called by: CSRCEND              (procedure in CREATESR.PRG)
*!                 : CSRCENDA             (procedure in CREATESR.PRG)
*!
*!           Calls: ECHELON.PRG
*!                : SETBROWS.PRG
*!                : ENDBROWS.PRG
*!                : DISPUNIT              (procedure in CREATESR.PRG)
*!
*!            Uses: ECHELONS.DBF
*!
*!        MDX files: ECHELONS.MDX
*!
*!*************************************************************
PROCEDURE adddata
SELECT 2
APPEND BLANK
REPLACE csrclist->src WITH temp->src
REPLACE csrclist->unit_desig WITH temp->unit_desig
REPLACE csrclist->strength WITH temp->strength
REPLACE csrclist->unitname WITH temp->unitname
REPLACE csrclist->qty WITH 1
* Try to guess echelon of this unit
DO echelon
* If no guess could be made
IF echelon=" "
   SELECT 25
   USE echelons
   DO setbrows
   SET MESSAGE TO "No echelon estimated. Please choose unit echelon, then Enter
to Select."
   ACTIVATE WINDOW browser
   BROWSE NOEDIT NOMENU NOAPPEND NODELETE FIELDS echelon
   DO endbrows
   DEACTIVATE WINDOW browser
   SELECT 2
   REPLACE echelon WITH echelons->echelon
ENDIF (echelon=" ")
DO dispunit
@ 21,5 SAY "Echelon: "+echelon
IF echelon="COMPANY"
   @ ROW(), COL() SAY " Level or Below"
ENDIF (echelon="COMPANY")
SET MESSAGE TO
SELECT 2

*: EOF: CREATESR.PRG
```

```
*:********************************************************************
*:
*:  Procedure file: D:\SURE\CREATETA.PRG
*:
*:   Procs & Fncts: STARTASK
*:                : CRET_PRO
*:                : NAMETA
*:                : ASSIGNUN
*:                : ASSIGNU2
*:                : UNITQTY
*:                : ENDTASK
*:                : ENDTASKA
*:                : DISPTASK
*:                : DISPUNI
*:
*:         Set by: TASK_PRO            (procedure in SURE.PRG)
*:
*:          Calls: STARTASK            (procedure in CREATETA.PRG)
*:               : CRET_PRO            (procedure in CREATETA.PRG)
*:
*:       Documented 04/16/94 at 11:01                SNAP!  version 5.00
*:********************************************************************
* createta.prg
STORE PROGRAM() TO progvar
* PRIVATE Variables
STORE "NO " TO yesnovar
* variable for taskorg name
STORE 0 TO countvar
STORE SPACE(60) TO namevar
STORE 1 TO statusvar
SET PROCEDURE TO insttask
DO startask
DEFINE POPUP cret_pop FROM 1,5
DEFINE BAR 1 OF cret_pop PROMPT "Name Task Organization"
DEFINE BAR 2 OF cret_pop PROMPT "Assign Units - Unit Name Order"
DEFINE BAR 3 OF cret_pop PROMPT "Assign Units - SRC Order"
DEFINE BAR 4 OF cret_pop PROMPT "Assign Unit Quantities"
DEFINE BAR 5 OF cret_pop PROMPT "Create from Selected Units"
DEFINE BAR 6 OF cret_pop PROMPT "Return to Main Menu"
ON SELECTION POPUP cret_pop DO cret_pro
ON BAR 1 OF cret_pop DO cret_bar1
ON BAR 2 OF cret_pop DO cret_bar2
ON BAR 3 OF cret_pop DO cret_bar3
ON BAR 4 OF cret_pop DO cret_bar4
ON BAR 5 OF cret_pop DO cret_bar5
ON BAR 6 OF cret_pop DO cret_bar6
DO WHILE PROMPT()<>"Return to Main Menu"
   ACTIVATE POPUP cret_pop
ENDDO (PROMPT()<>"Return to Main Menu")
*!********************************************************************
*!
*!        Procedure: CRET_PRO
*!
*!        Called by: CREATETA.PRG
*!
*!            Calls: NAMETA              (procedure in CREATETA.PRG)
*!                 : ASSIGNUN            (procedure in CREATETA.PRG)
*!                 : ASSIGNU2            (procedure in CREATETA.PRG)
*!                 : UNITQTY             (procedure in CREATETA.PRG)
*!                 : ENDTASKA            (procedure in CREATETA.PRG)
```

111

```
*!                    : ENDTASK            (procedure in CREATETA.PRG)
*!
*!*****************************************************************
PROCEDURE cret_pro
DO CASE
CASE BAR()=1
    DO nameta
CASE BAR()=2
    DO assignun
CASE BAR()=3
    DO assignu2
CASE BAR()=4
    DO unitqty
CASE BAR()=5
    DO endtaska
CASE BAR()=6
    DO endtask
ENDCASE
* Procedures for CreT_pop
*!*****************************************************************
*!
*!       Procedure: NAMETA
*!
*!       Called by: CRET_PRO            (procedure in CREATETA.PRG)
*!
*!          Calls: DISPTASK            (procedure in CREATETA.PRG)
*!
*!*****************************************************************
PROCEDURE nameta
SET MESSAGE TO "Assign Name for this Task Organization"
ACTIVATE WINDOW medium
DO WHILE .T.
    @ 1,20 SAY "Enter Task Organization Name: "
    @ 3,7 GET namevar
    READ
    IF namevar=" "
        ? CHR(7)
        CLEAR
        @ 1,10 SAY "You can't have a Task Organization without a Name!"
        @ 3,10
        WAIT
        SET MESSAGE TO
        DEACTIVATE WINDOW medium
        EXIT
    ELSE
        SELECT 1
        SET ORDER TO taskorgnam
        SEEK namevar
        IF FOUND()
            ? CHR(7)
            CLEAR
            @ 1,10 SAY "Duplicate Name!"
            @ 3,10
            WAIT
            CLEAR
        ELSE
            CLEAR
            DEACTIVATE WINDOW medium
            DO disptask
            statusvar=2
```

112

```
            SET MESSAGE TO
            EXIT
        ENDIF (FOUND())
    ENDIF (namevar=" ")
ENDDO (.T.)
*!********************************************************************
*!
*!          Procedure: ASSIGNUN
*!
*!          Called by: CRET_PRO              (procedure in CREATETA.PRG)
*!                    : EDIT_PRO              (procedure in EDITTASK.PRG)
*!
*!              Calls: DISPTASK               (procedure in CREATETA.PRG)
*!
*!               Uses: TEMP.DBF
*!                   : TTEMP.DBF
*!
*!********************************************************************
PROCEDURE assignun
IF statusvar=1
    ACTIVATE WINDOW small
    ? CHR(7)
    @ 1,2 SAY "Assign a Task Organization First!"
    @ 2,2 SAY "Press any key..."
    WAIT ""
    DEACTIVATE WINDOW small
    RETURN
ELSE
    SET MESSAGE TO "Enter Y for units in this Task Organization, then Ctrl-W"
    ACTIVATE WINDOW browser
    SELECT 2
    REPLACE ALL taskorg WITH .n. FOR taskorg
    GO TOP
    BROWSE FIELDS taskorg,src,unitname NOAPPEND NODELETE NOMENU FREEZE taskorg
    DEACTIVATE WINDOW browser
    SET MESSAGE TO
ENDIF (statusvar=1)
*!********************************************************************
*!
*!          Procedure: ASSIGNU2
*!
*!          Called by: CRET_PRO              (procedure in CREATETA.PRG)
*!                    : EDIT_PRO              (procedure in EDITTASK.PRG)
*!
*!              Calls: DISPTASK               (procedure in CREATETA.PRG)
*!
*!               Uses: TEMP.DBF
*!                   : TTEMP.DBF
*!
*!********************************************************************
PROCEDURE assignu2
IF statusvar=1
    ACTIVATE WINDOW small
    ? CHR(7)
    @ 1,2 SAY "Assign a Task Organization First!"
    @ 2,2 SAY "Press any key..."
    WAIT ""
    DEACTIVATE WINDOW small
    RETURN
ELSE
```

113

```
        SET MESSAGE TO "Enter Y for units in this Task Organization, then Ctrl-W"
        ACTIVATE WINDOW browser
        SELECT 2
        REPLACE ALL taskorg WITH .n. FOR taskorg
        SET ORDER TO src
        GO TOP
        BROWSE FIELDS taskorg,src,unitname NOAPPEND NODELETE NOMENU FREEZE taskorg
        DEACTIVATE WINDOW browser
        SET MESSAGE TO
ENDIF (statusvar=1)
*!*******************************************************************
*!
*!       Procedure: UNITQTY
*!
*!       Called by: CRET_PRO              (procedure in CREATETA.PRG)
*!
*!*******************************************************************
PROCEDURE unitqty
IF statusvar=1
        ACTIVATE WINDOW small
        ? CHR(7)
        @ 1,2 SAY "Assign a Task Organization First!"
        @ 2,2 SAY "Press any key..."
        WAIT ""
        DEACTIVATE WINDOW small
        RETURN
ELSE
        SET MESSAGE TO "Enter quantity 1 or more of units in this Task Organization,
then Ctrl-W"
        ACTIVATE WINDOW browser
        SELECT 2
        REPLACE ALL qty WITH 0 FOR qty>0
        GO TOP
        SET FILTER TO taskorg
        GO TOP
        BROWSE FIELDS qty,src,unitname NOAPPEND NODELETE NOMENU FREEZE qty
        SET FILTER TO
        DEACTIVATE WINDOW browser
ENDIF (statusvar=1)
*!*******************************************************************
*!
*!       Procedure: ENDTASK
*!
*!       Called by: CRET_PRO              (procedure in CREATETA.PRG)
*!
*!             Uses: TEMP.DBF
*!                 : TTEMP.DBF
*!
*!*******************************************************************
PROCEDURE endtask
IF statusvar=2
        ACTIVATE WINDOW small
        @ 1,1 SAY "Create Task Organization from"
        @ 2,2 SAY "Selected Units? "
        @ ROW(), COL() GET yesnovar PICTURE "!!!"
        READ
        DEACTIVATE WINDOW small
        IF yesnovar="YES"
           SELECT 1
           COPY STRUCTURE TO temp
```

114

```
                SELECT 2
                COPY TO ttemp FOR taskorg
                * Check to see some units had been selected
                SELECT 16
                USE ttemp
                IF RECCOUNT()=0
                    SELECT 16
                    USE
                    ACTIVATE WINDOW small
                    ? CHR(7)
                    @ 1,2 SAY "No Units Assigned,"
                    @ 2,2 SAY "No Task Force Created!"
                    @ 3,2 SAY "Press any key..."
                    WAIT ""
                    DEACTIVATE WINDOW small
                ELSE
                    REPLACE ALL qty WITH 1 FOR qty<1
                    SELECT 16
                    USE
                    SELECT 11
                    USE temp
                    APPEND FROM ttemp
                    REPLACE ALL taskorgnam WITH namevar
                    *REPLACE ALL qty WITH 1 FOR qty<1
                    SELECT 11
                    USE
                    SELECT 1
                    APPEND FROM temp
                ENDIF (RECCOUNT()=0)
                SELECT 2
                REPLACE ALL taskorg WITH .n. FOR taskorg
            ENDIF (yesnovar="YES")
        ENDIF (statusvar=2)
        CLOSE DATA
        CLEAR
        DEACTIVATE POPUP
        *!****************************************************************
        *!
        *!       Procedure: ENDTASKA
        *!
        *!       Called by: CRET_PRO              (procedure in CREATETA.PRG)
        *!
        *!            Uses: TEMP.DBF
        *!                : TTEMP.DBF
        *!
        *!****************************************************************
        PROCEDURE endtaska
        IF statusvar=1
            ACTIVATE WINDOW small
            ? CHR(7)
            @ 1,2 SAY "Assign a Task Organization First!"
            @ 2,2 SAY "Press any key..."
            WAIT ""
            DEACTIVATE WINDOW small
            RETURN
        ENDIF (statusvar=1)
        SELECT 1
        COPY STRUCTURE TO temp
        SELECT 2
        COPY TO ttemp FOR taskorg
```

115

```
* Check to see some units had been selected
SELECT 16
USE ttemp
IF RECCOUNT()=0
    SELECT 16
    USE
    ACTIVATE WINDOW small
    ? CHR(7)
    @ 1,2 SAY "Assign some units First!"
    @ 2,2 SAY "Press any key..."
    WAIT ""
    DEACTIVATE WINDOW small
    RETURN
ENDIF (RECCOUNT()=0)
REPLACE ALL qty WITH 1 FOR qty<1
SELECT 16
USE
SELECT 11
USE temp
APPEND FROM ttemp
REPLACE ALL taskorgnam WITH namevar
SELECT 11
USE
SELECT 1
APPEND FROM temp
statusvar=1
SET MESSAGE TO "Give new Task Organization a Name"
*!****************************************************************
*!
*!        Procedure: STARTASK
*!
*!        Called by: CREATETA.PRG
*!                 : CREATETC.PRG
*!
*!            Calls: STARTHLP.PRG
*!
*!             Uses: TASKORG.DBF
*!                 : CSRCLIST.DBF
*!
*!        MDX files: TASKORG.MDX
*!                 : CSRCLIST.MDX
*!
*!****************************************************************
PROCEDURE startask
CLEAR
CLOSE DATA
DO starthlp
SELECT 1
USE taskorg
SELECT 2
USE csrclist
SELECT 1
*!****************************************************************
*!
*!        Procedure: DISPTASK
*!
*!        Called by: NAMETA                (procedure in CREATETA.PRG)
*!                 : ASSIGNUN              (procedure in CREATETA.PRG)
*!                 : ASSIGNU2              (procedure in CREATETA.PRG)
*!                 : FINDTA                (procedure in CREATETC.PRG)
```

116

```
*!                      : ASSIGNNA          (procedure in CREATETC.PRG)
*!                      : VIEWTASK          (procedure in EDITTASK.PRG)
*!                      : VIEWTAS2          (procedure in EDITTASK.PRG)
*!                      : UNASSIGU          (procedure in EDITTASK.PRG)
*!                      : UNASSIG2          (procedure in EDITTASK.PRG)
*!
*!*************************************************************************
PROCEDURE disptask
@ 17,2 TO 17,78
@ 21,2 SAY "Task Organization:   "
@ ROW(),COL() SAY namevar
*!*************************************************************************
*!
*!       Procedure: DISPUNI
*!
*!*************************************************************************
PROCEDURE dispuni
@ 19,2 SAY "       PARENT UNIT:   "
@ ROW(),COL() SAY unitvar
RETURN
*: EOF: CREATETA.PRG
```

117

```
*:********************************************************************
*:
*:  Procedure file: D:\SURE\CREATETC.PRG
*:
*:    Procs & Fncts: COPT_PRO
*:                 : FINDTA
*:                 : ASSIGNNA
*:                 : ENDCOPT
*:                 : ENDCOPTA
*:
*:         Set by: TASK_PRO              (procedure in SURE.PRG)
*:
*:          Calls: STARTASK              (procedure in CREATETA.PRG)
*:               : COPT_PRO              (procedure in CREATETC.PRG)
*:
*:        Documented 04/16/94 at 11:01            SNAP!  version 5.00
*:********************************************************************
* createta.prg
STORE PROGRAM() TO progvar
* PRIVATE Variables
STORE "NO " TO yesnovar
CLEAR
* variable for taskorg name
STORE 0 TO countvar
STORE SPACE(60) TO namevar
STORE SPACE(60) TO namevar2
STORE 1 TO statusvar
SET PROCEDURE TO insttask
DO startask
DEFINE POPUP copt_pop FROM 1,5
DEFINE BAR 1 OF copt_pop PROMPT "Select Task Organization to Copy"
DEFINE BAR 2 OF copt_pop PROMPT "Assign New Task Organization Name"
DEFINE BAR 3 OF copt_pop PROMPT "Save and Copy Another"
DEFINE BAR 4 OF copt_pop PROMPT "Return to Main Menu"
ON SELECTION POPUP copt_pop DO copt_pro
ON BAR 1 OF copt_pop DO copt_bar1
ON BAR 2 OF copt_pop DO copt_bar2
ON BAR 3 OF copt_pop DO copt_bar3
ON BAR 4 OF copt_pop DO copt_bar4
DO WHILE PROMPT()<>"Return to Main Menu"
   ACTIVATE POPUP copt_pop
ENDDO (PROMPT()<>"Return to Main Menu")
*!********************************************************************
*!
*!      Procedure: COPT_PRO
*!
*!      Called by: CREATETC.PRG
*!
*!          Calls: FINDTA                (procedure in CREATETC.PRG)
*!               : ASSIGNNA              (procedure in CREATETC.PRG)
*!               : ENDCOPT               (procedure in CREATETC.PRG)
*!               : ENDCOPTA              (procedure in CREATETC.PRG)
*!
*!********************************************************************
PROCEDURE copt_pro
DO CASE
CASE BAR()=1
   DO findta
CASE BAR()=2
   DO assignna
```

```
CASE BAR()=4
   DO endcopt .
CASE BAR()=3
   DO endcopta
ENDCASE
* Open files
*!***********************************************************************
*!
*!        Procedure: STARTASK
*!
*!        Called by: CREATETA.PRG
*!                 : CREATETC.PRG
*!
*!            Calls: STARTHLP.PRG
*!
*!             Uses: TASKORG.DBF
*!                 : CSRCLIST.DBF
*!
*!        MDX files: TASKORG.MDX
*!                 : CSRCLIST.MDX
*!
*!***********************************************************************
PROCEDURE startask
CLEAR
DO starthlp
SELECT 1
USE taskorg
* Select Task Organization
*!***********************************************************************
*!
*!        Procedure: FINDTA
*!
*!        Called by: EDITTASK.PRG
*!                 : COPT_PRO          (procedure in CREATETC.PRG)
*!                 : EDIT_PRO          (procedure in EDITTASK.PRG)
*!                 : DELT_PRO          (procedure in DELTASK.PRG)
*!
*!            Calls: SETBROWS.PRG
*!                 : ENDBROWS.PRG
*!                 : DISPTASK          (procedure in CREATETA.PRG) .
*!
*!***********************************************************************
PROCEDURE findta
SELECT 1
SET ORDER TO uniqueta              && only view each task order name once
DO setbrows
BROWSE FIELDS taskorgnam NOMENU NOEDIT NOAPPEND NODELETE
DO endbrows
STORE taskorgnam TO namevar                && Will replace with new name
STORE taskorgnam TO namevar2               && Needed to copy old Task Org
DO disptask
SET ORDER TO taskorgnam
statusvar=2
* Assign new Task Org name
*!***********************************************************************
*!
*!        Procedure: ASSIGNNA
*!
*!        Called by: COPT_PRO              (procedure in CREATETC.PRG)
*!
```

119

```
*!          Calls: DISPTASK          (procedure in CREATETA.PRG)
*!
*!**********************************************************************
PROCEDURE assignna
IF statusvar=1
    ACTIVATE WINDOW small
    ? CHR(7)
    @ 1,2 SAY "Select Task Organization First!"
    @ 2,2 SAY "Press any key..."
    WAIT ""
    DEACTIVATE WINDOW small
    RETURN
ENDIF (statusvar=1)
SET MESSAGE TO "Assign Name for this Task Organization"
ACTIVATE WINDOW medium
DO WHILE .T.
    @ 1,20 SAY "Enter Task Organization Name: "
    @ 3,7 GET namevar
    READ
    IF namevar=" "
        ? CHR(7)
        CLEAR
        @ 1,10 SAY "You can't have a Task Organization without a Name!"
        @ 3,10
        WAIT
        SET MESSAGE TO
        DEACTIVATE WINDOW medium
        EXIT
    ELSE
        SELECT 1
        SET ORDER TO taskorgnam
        SEEK namevar
        IF FOUND()
            ? CHR(7)
            CLEAR
            @ 1,10 SAY "Duplicate Name!"
            @ 3,10
            WAIT
            CLEAR
        ELSE
            CLEAR
            DEACTIVATE WINDOW medium
            DO disptask
            SET MESSAGE TO
            statusvar=3
            EXIT
        ENDIF (FOUND())
    ENDIF (namevar=" ")
ENDDO (.T.)
* DONE
*!**********************************************************************
*!
*!      Procedure: ENDCOPT
*!
*!      Called by: COPT_PRO          (procedure in CREATETC.PRG)
*!
*!           Uses: TEMP.DBF
*!
*!**********************************************************************
PROCEDURE endcopt
```

```
IF statusvar=3                  && All required actions completed
   ACTIVATE WINDOW small
   @ 1,1 SAY "Create Task Organization from"
   @ 2,2 SAY "Copied Units? "
   @ ROW(), COL() GET yesnovar PICTURE "!!!"
   READ
   DEACTIVATE WINDOW small
   IF yesnovar="YES"
      SELECT 1
      SET ORDER TO taskorgnam
      COPY TO temp FOR taskorgnam=namevar2
      SELECT 11
      USE temp
      REPLACE ALL taskorgnam WITH namevar
      SELECT 11
      USE
      SELECT 1
      APPEND FROM temp
      SELECT 11
      USE temp
      ZAP
   ENDIF (yesnovar="YES")
ENDIF (statusvar=3              && All required actions completed)
CLOSE DATA
DEACTIVATE POPUP
* DONE - and copy another
*!*****************************************************************
*!
*!      Procedure: ENDCOPTA
*!
*!      Called by: COPT_PRO           (procedure in CREATETC.PRG)
*!
*!          Uses: TEMP.DBF
*!
*!*****************************************************************
PROCEDURE endcopta
IF statusvar<>3                 && All required actions not completed
   ACTIVATE WINDOW small
   ? CHR(7)
   @ 1,2 SAY "No Task Force Copied.."
   @ 2,2 SAY "No Task Force Created!"
   @ 3,2 SAY "Press any key..."
   WAIT ""
   DEACTIVATE WINDOW small
ELSE
   SELECT 1
   SET ORDER TO taskorgnam
   COPY TO temp FOR taskorgnam=namevar2
   SELECT 11
   USE temp
   REPLACE ALL taskorgnam WITH namevar
   SELECT 11
   USE
   SELECT 1
   APPEND FROM temp
   SELECT 11
   USE temp
   ZAP
   statusvar=1
ENDIF (statusvar<>3            && All required actions not completed)
```

```
*!*********************************************************************
*!
*!          Procedure: DISPTASK
*!
*!          Called by: NAMETA                (procedure in CREATETA.PRG)
*!                    : ASSIGNUN              (procedure in CREATETA.PRG)
*!                    : ASSIGNU2              (procedure in CREATETA.PRG)
*!                    : FINDTA                (procedure in CREATETC.PRG)
*!                    : ASSIGNNA              (procedure in CREATETC.PRG)
*!                    : VIEWTASK              (procedure in EDITTASK.PRG)
*!                    : VIEWTAS2              (procedure in EDITTASK.PRG)
*!                    : UNASSIGU              (procedure in EDITTASK.PRG)
*!                    : UNASSIG2              (procedure in EDITTASK.PRG)
*!
*!*********************************************************************
PROCEDURE disptask
@ 17,2 TO 17,78
@ 21,2 SAY "Task Organization:   "
@ ROW(),COL() SAY namevar

*: EOF: CREATETC.PRG
```

```
*:****************************************************************
*:
*:  Procedure file: D:\SURE\CRECUSPR.PRG
*:
*:   Procs & Fncts: CRCP_PRO
*:                : NEWCUSPR
*:                : VIEWCUSP
*:                : EDITCUSP
*:                : ENDCUSPR
*:                : DELCUSPR
*:                : DISPPRO
*:
*:          Set by: PROF_PRO           (procedure in SURE.PRG)
*:
*:           Calls: STARTPRO           (procedure in CREATEPR.PRG)
*:                : CRCP_PRO           (procedure in CRECUSPR.PRG)
*:
*:        Documented 04/16/94 at 11:01          SNAP!  version 5.00
*:****************************************************************
* createpro.prg
CLEAR
STORE PROGRAM() TO progvar
SET PROCEDURE TO instprof
DO startpro
DEFINE POPUP crcp_pop FROM 1,14
DEFINE BAR 1 OF crcp_pop PROMPT "Create Custom Area Profile"
DEFINE BAR 2 OF crcp_pop PROMPT "View Area Profiles"
DEFINE BAR 3 OF crcp_pop PROMPT "Edit Custom Area Profile"
DEFINE BAR 4 OF crcp_pop PROMPT "Return to Main Menu"
DEFINE BAR 6 OF crcp_pop PROMPT "Delete Custom Area Profile"
ON SELECTION POPUP crcp_pop DO crcp_pro
ON BAR 1 OF crcp_pop DO crcp_bar1
ON BAR 2 OF crcp_pop DO crcp_bar2
ON BAR 3 OF crcp_pop DO crcp_bar3
ON BAR 4 OF crcp_pop DO crcp_bar4
ON BAR 6 OF crcp_pop DO crcp_bar6
DO WHILE PROMPT()<>"Return to Main Menu"
   ACTIVATE POPUP crcp_pop
ENDDO (PROMPT()<>"Return to Main Menu")
*!****************************************************************
*!
*!      Procedure: CRCP_PRO
*!
*!      Called by: CRECUSPR.PRG
*!
*!          Calls: NEWCUSPR            (procedure in CRECUSPR.PRG)
*!               : VIEWCUSP            (procedure in CRECUSPR.PRG)
*!               : EDITCUSP            (procedure in CRECUSPR.PRG)
*!               : ENDCUSPR            (procedure in CRECUSPR.PRG)
*!               : DELCUSPR            (procedure in CRECUSPR.PRG)
*!
*!****************************************************************
PROCEDURE crcp_pro
DO CASE
CASE BAR()=1
   DO newcuspr
CASE BAR()=2
   DO viewcusp
CASE BAR()=3
   DO editcusp
```

123

```
CASE BAR()=4
   DO endcuspr
CASE BAR()=6
   DO delcuspr
ENDCASE
* Procedures for crcp_pop
* CASE BAR() = 1
*!**********************************************************************
*!
*!         Procedure: NEWCUSPR
*!
*!         Called by: CRCP_PRO            (procedure in CRECUSPR.PRG)
*!
*!             Calls: SETBROWS.PRG
*!                  : ENDBROWS.PRG
*!
*!              Uses: TEMP.DBF
*!
*!**********************************************************************
PROCEDURE newcuspr
SELECT 7
COPY STRUCTURE TO temp
SELECT 11
USE temp
APPEND BLANK
SELECT 7
SET FILTER TO          .
ACTIVATE WINDOW browser
DO setbrows
SET MESSAGE TO " or  to move, Enter to Select Profile to base Custom Settings
On"
BROWSE NOAPPEND NOEDIT NODELETE FIELDS custom, profile COMPRESS
DO endbrows
DEACTIVATE WINDOW browser
* Replace temp.dbf with selected Profile data
REPLACE temp->ae_hrs WITH profiles->ae_hrs
REPLACE temp->av_hrs WITH profiles->av_hrs
REPLACE temp->ce_hrs WITH profiles->ce_hrs
REPLACE temp->gn_hrs WITH profiles->gn_hrs
REPLACE temp->hg_hrs WITH profiles->hg_hrs
REPLACE temp->mh_hrs WITH profiles->mh_hrs
REPLACE temp->ov_hrs WITH profiles->ov_hrs
REPLACE temp->sg_hrs WITH profiles->sg_hrs
REPLACE temp->sv_hrs WITH profiles->sv_hrs
REPLACE temp->wv_kms WITH profiles->wv_kms
REPLACE temp->tv_ti_hrs WITH profiles->tv_ti_hrs
REPLACE temp->tv_cc_hrs WITH profiles->tv_cc_hrs
REPLACE temp->tv_sr_hrs WITH profiles->tv_sr_hrs
REPLACE temp->custom WITH .y.
SELECT 11
KEYBOARD CHR(127)
SET MESSAGE TO "Enter Custom Profile Name and Data, then Ctrl-W to Save"
CLEAR
@ 1,2 SAY " Enter Custom Profile Name "
@ ROW(),COL() GET profile
@ ROW(),COL() SAY " Based on: "+profiles->profile
@ 2,2 SAY "              If no Profile Name is entered, data WILL NOT be
saved."
@ 4,2 SAY "                    Amphibious Equipment "
@ ROW(), COL() GET ae_hrs
```

124

```
@ ROW(), COL() SAY " hours/day"
@ 5,2 SAY "                                Construction "
@ ROW(), COL() GET ce_hrs
@ ROW(), COL() SAY " hours/day"
@ 6,2 SAY "                                 Generators "
@ ROW(), COL() GET gn_hrs
@ ROW(), COL() SAY " hours/day"
@ 7,2 SAY "                         Heating Equipment "
@ ROW(), COL() GET hg_hrs
@ ROW(), COL() SAY " hours/day"
@ 8,2 SAY "            Materials Handling Equipment "
@ ROW(), COL() GET mh_hrs
@ ROW(), COL() SAY " hours/day"
@ 9,2 SAY "  Stationary Equipment - Miscellaneous "
@ ROW(), COL() GET sg_hrs
@ ROW(), COL() SAY " hours/day"
@ 10,2 SAY "Stationary Equipment - Vehicle Mounted "
@ ROW(), COL() GET sv_hrs
@ ROW(), COL() SAY " hours/day"
@ 12,2 SAY "                   Tracked Vehicles - Idle "
@ ROW(), COL() GET tv_ti_hrs
@ ROW(), COL() SAY " hours/day"
@ 13,2 SAY "      Tracked Vehicles - Cross Country "
@ ROW(), COL() GET tv_cc_hrs
@ ROW(), COL() SAY " hours/day"
@ 14,2 SAY "    Tracked Vehicles - Secondary Roads "
@ ROW(), COL() GET tv_sr_hrs
@ ROW(), COL() SAY " hours/day"
@ 16,2 SAY "                         Wheeled Vehicles "
@ ROW(), COL() GET wv_kms
@ ROW(), COL() SAY " kilometers/day"
@ 17,2 SAY "                           Other Vehicles "
@ ROW(), COL() GET ov_hrs
@ ROW(), COL() SAY " hours/day"
@ 19,2 SAY "                               Aviation "
@ ROW(), COL() GET av_hrs
@ ROW(), COL() SAY " hours/day"
READ
CLEAR
SET MESSAGE TO
* Make sure a name was added
IF profile=" "
   ACTIVATE WINDOW small
   ? CHR(7)
   @ 1,2 SAY "Set NOT Named   "
   @ 2,2 SAY "Set NOT added!      "
   @ 3,3 SAY "Press any key..."
   WAIT ""
   SELECT 11
   ZAP
   SELECT 11
   USE
   DEACTIVATE WINDOW small
   CLEAR
   RETURN
ENDIF (profile=" ")
* Make sure name is not a duplicate
SELECT 7
SEEK temp->profile
IF .NOT. EOF()
```

```
   ACTIVATE WINDOW small
   ? CHR(7)
   @ 1,2 SAY "Duplicate Profile Name"
   @ 2,2 SAY "Profile NOT added!"
   @ 3,3 SAY "Press any key..."
   WAIT ""
   SELECT 11
   ZAP
   SELECT 11
   USE
   DEACTIVATE WINDOW small
   CLEAR
   RETURN
ENDIF (.NOT. EOF())
SELECT 11
USE
SELECT 7
APPEND FROM temp
* CASE BAR() = 2
*!*********************************************************************
*!
*!       Procedure: VIEWCUSP
*!
*!       Called by: CRCP_PRO              (procedure in CRECUSPR.PRG)
*!
*!           Calls: SETBROWS.PRG
*!                : ENDBROWS.PRG
*!                : DISPPRO               (procedure in CRECUSPR.PRG)
*!
*!*********************************************************************
PROCEDURE viewcusp
SELECT 7
ACTIVATE WINDOW browser
GO TOP
DO setbrows
BROWSE NOAPPEND NOEDIT NODELETE FIELDS custom, profile COMPRESS
DO endbrows
DEACTIVATE WINDOW browser
CLEAR
DO disppro
WAIT ""
@ 23, 2 SAY "                                               "
CLEAR
* CASE BAR() = 3
*!*********************************************************************
*!
*!       Procedure: EDITCUSP
*!
*!       Called by: CRCP_PRO              (procedure in CRECUSPR.PRG)
*!
*!           Calls: SETBROWS.PRG
*!                : ENDBROWS.PRG
*!
*!*********************************************************************
PROCEDURE editcusp
SELECT 7
SET FILTER TO custom
GO TOP
IF EOF()
   ACTIVATE WINDOW small
```

126

```
      ? CHR(7)
      @ 1,2 SAY "No Custom Profiles!"
      @ 3,3 SAY "Press any key..."
      WAIT ""
      DEACTIVATE WINDOW small
      SET FILTER TO
      CLEAR
      RETURN
  ENDIF (EOF())
  DO setbrows
  BROWSE NOAPPEND NOEDIT NODELETE FIELDS custom, profile COMPRESS
  DO endbrows
  @ 1,2 SAY "         Custom Profile Name "
  @ ROW(),COL() SAY profile
  @ 4,2 SAY "                      Amphibious Equipment "
  @ ROW(), COL() GET ae_hrs
  @ ROW(), COL() SAY " hours/day"
  @ 5,2 SAY "                            Construction "
  @ ROW(), COL() GET ce_hrs
  @ ROW(), COL() SAY " hours/day"
  @ 6,2 SAY "                             Generators "
  @ ROW(), COL() GET gn_hrs
  @ ROW(), COL() SAY " hours/day"
  @ 7,2 SAY "                     Heating Equipment "
  @ ROW(), COL() GET hg_hrs
  @ ROW(), COL() SAY " hours/day"
  @ 8,2 SAY "           Materials Handling Equipment "
  @ ROW(), COL() GET mh_hrs
  @ ROW(), COL() SAY " hours/day"
  @ 9,2 SAY "  Stationary Equipment - Miscellaneous "
  @ ROW(), COL() GET sg_hrs
  @ ROW(), COL() SAY " hours/day"
  @ 10,2 SAY "Stationary Equipment - Vehicle Mounted "
  @ ROW(), COL() GET sv_hrs
  @ ROW(), COL() SAY " hours/day"
  @ 12,2 SAY "               Tracked Vehicles - Idle "
  @ ROW(), COL() GET tv_ti_hrs
  @ ROW(), COL() SAY " hours/day"
  @ 13,2 SAY "      Tracked Vehicles - Cross Country "
  @ ROW(), COL() GET tv_cc_hrs
  @ ROW(), COL() SAY " hours/day"
  @ 14,2 SAY "    Tracked Vehicles - Secondary Roads "
  @ ROW(), COL() GET tv_sr_hrs
  @ ROW(), COL() SAY " hours/day"
  @ 16,2 SAY "                       Wheeled Vehicles "
  @ ROW(), COL() GET wv_kms
  @ ROW(), COL() SAY " kilometers/day"
  @ 17,2 SAY "                         Other Vehicles "
  @ ROW(), COL() GET ov_hrs
  @ ROW(), COL() SAY " hours/day"
  @ 19,2 SAY "                             Aviation "
  @ ROW(), COL() GET av_hrs
  @ ROW(), COL() SAY " hours/day"
  @ 21,2 SAY "              Enter Changes, then CTRL-W to Return to Menu"
  READ
  CLEAR
  SET MESSAGE TO
  SET FILTER TO
  RETURN
  * CASE BAR() = 4
```

```
*!*****************************************************************
*!
*!          Procedure: ENDCUSPR
*!
*!          Called by: CRCP_PRO            (procedure in CRECUSPR.PRG)
*!
*!*****************************************************************
PROCEDURE endcuspr
CLOSE DATA
DEACTIVATE POPUP
* CASE BAR() = 6
*!*****************************************************************
*!
*!          Procedure: DELCUSPR
*!
*!          Called by: CRCP_PRO            (procedure in CRECUSPR.PRG)
*!
*!              Calls: SETBROWS.PRG
*!                   : ENDBROWS.PRG
*!                   : DISPPRO             (procedure in CRECUSPR.PRG)
*!
*!*****************************************************************
PROCEDURE delcuspr
SELECT 7
SET FILTER TO custom
GO TOP
IF EOF()
   ACTIVATE WINDOW small
   ? CHR(7)
   @ 1,2 SAY "No Custom Profiles!"
   @ 3,3 SAY "Press any key..."
   WAIT ""
   DEACTIVATE WINDOW small
   SET FILTER TO
   CLEAR
   RETURN
ENDIF (EOF())
ACTIVATE WINDOW browser
DO setbrows
BROWSE NOAPPEND NOEDIT NODELETE FIELDS custom, profile COMPRESS
DO endbrows
DEACTIVATE WINDOW browser
CLEAR
DO disppro
STORE "NO " TO yesnovar
ACTIVATE WINDOW small
@ 1,1 SAY "Enter YES to Delete this Profile "
@ ROW(), COL() GET yesnovar PICTURE "!!!"
READ
DEACTIVATE WINDOW small
IF yesnovar="YES"
   SELECT 4
   GO TOP
   SEEK profiles->profile
   IF FOUND()
      ? CHR(7)
      ACTIVATE WINDOW browser
      TEXT

           This Profile is used in a Parameter Set.  You can not delete it!
```

128

```
                    Press any key to continue....

        ENDTEXT
        WAIT ""
        DEACTIVATE WINDOW browser
        SELECT 7
        SET FILTER TO
        CLEAR
        RETURN
    ENDIF (FOUND())
ELSE
    SET FILTER TO
    CLEAR
    RETURN
ENDIF (yesnovar="YES")
SELECT 7
DELETE
SET FILTER TO
PACK
CLEAR
RETURN
*!*********************************************************************
*!
*!        Procedure: STARTPRO
*!
*!        Called by: CREATEPR.PRG
*!                 .: CRECUSPR.PRG
*!
*!            Calls: STARTHLP.PRG
*!
*!             Uses: MEDICAL.DBF
*!                 : WATER.DBF
*!                 : CLASS_I.DBF
*!                 : POSTURE.DBF
*!                 : INTENSTY.DBF
*!                 : PROFILES.DBF
*!                 : CPROFILE.DBF
*!
*!        MDX files: MEDICAL.MDX
*!                 : WATER.MDX
*!                 : CLASS_I.MDX
*!                 : INTENSTY.MDX
*!                 : PROFILES.MDX
*!                 : CPROFILE.MDX
*!
*!*********************************************************************
PROCEDURE startpro
DO starthlp
SELECT 4
USE cprofile
SET ORDER TO profile
SELECT 7
USE profiles ORDER profile
STORE SPACE(60) TO namevar
*!*********************************************************************
*!
*!        Procedure: DISPPRO
*!
*!        Called by: VIEWCUSP              (procedure in CRECUSPR.PRG)
*!                 : DELCUSPR              (procedure in CRECUSPR.PRG)
```

129

```
*!
*!*******************************************************************
PROCEDURE disppro
@ 1,2 SAY "                        Profile Name "
@ ROW(),COL() SAY profile
@ 4,2 SAY "                            Amphibious Equipment "
@ ROW(), COL() SAY ae_hrs
@ ROW(), COL() SAY " hours/day"
@ 5,2 SAY "                                    Construction "
@ ROW(), COL() SAY ce_hrs
@ ROW(), COL() SAY " hours/day"
@ 6,2 SAY "                                    Generators "
@ ROW(), COL() SAY gn_hrs
@ ROW(), COL() SAY " hours/day"
@ 7,2 SAY "                          Heating Equipment "
@ ROW(), COL() SAY hg_hrs
@ ROW(), COL() SAY " hours/day"
@ 8,2 SAY "             Materials Handling Equipment "
@ ROW(), COL() SAY mh_hrs
@ ROW(), COL() SAY " hours/day"
@ 9,2 SAY "  Stationary Equipment - Miscellaneous "
@ ROW(), COL() SAY sg_hrs
@ ROW(), COL() SAY " hours/day"
@ 10,2 SAY "Stationary Equipment - Vehicle Mounted "
@ ROW(), COL() SAY sv_hrs
@ ROW(), COL() SAY " hours/day"
@ 12,2 SAY "                   Tracked Vehicles - Idle "
@ ROW(), COL() SAY tv_ti_hrs
@ ROW(), COL() SAY " hours/day"
@ 13,2 SAY "        Tracked Vehicles - Cross Country "
@ ROW(), COL() SAY tv_cc_hrs
@ ROW(), COL() SAY " hours/day"
@ 14,2 SAY "    Tracked Vehicles - Secondary Roads "
@ ROW(), COL() SAY tv_sr_hrs
@ ROW(), COL() SAY " hours/day"
@ 16,2 SAY "                           Wheeled Vehicles "
@ ROW(), COL() SAY wv_kms
@ ROW(), COL() SAY " kilometers/day"
@ 17,2 SAY "                              Other Vehicles "
@ ROW(), COL() SAY ov_hrs
@ ROW(), COL() SAY " hours/day"
@ 19,2 SAY "                                    Aviation "
@ ROW(), COL() SAY av_hrs
@ ROW(), COL() SAY " hours/day"
@ 23, 2 SAY "                        Press any Key to Continue..."

*: EOF: CRECUSPR.PRG
```

```
*:*******************************************************************
*:
*: Procedure file: D:\SURE\CUSTUNIT.PRG
*:
*:  Procs & Fncts: CUST_PRO
*:               : UNITDATA
*:
*:        Set by: SRC_PRO              (procedure in SURE.PRG)
*:
*:         Calls: STARTUNI            (procedure in EDITUNIT.PRG)
*:              : CUST_PRO            (procedure in CUSTUNIT.PRG)
*:
*:        Documented 04/16/94 at 11:01              SNAP!  version 5.00
*:*******************************************************************
* editunit.prg
* creates master/subordinate unit relationships.
* PRIVATE Variables
** statusvar controls user attempts to do actions out of sequence
STORE PROGRAM() TO progvar
STORE 1 TO statusvar
SET PROCEDURE TO instunit
DO startuni
SET MESSAGE TO "Create Custom Unit"
DEFINE POPUP cust_pop FROM 1,0
DEFINE BAR 1 OF cust_pop PROMPT "Enter Unit Data"
DEFINE BAR 2 OF cust_pop PROMPT "View/Change LIN Quantities"
DEFINE BAR 3 OF cust_pop PROMPT "Add LINs"
DEFINE BAR 4 OF cust_pop PROMPT "Delete LIN"
DEFINE BAR 5 OF cust_pop PROMPT "Change Unit Echelon"
DEFINE BAR 6 OF cust_pop PROMPT "Return to Main Menu"
ON SELECTION POPUP cust_pop DO cust_pro
ON BAR 1 OF cust_pop DO cust_bar1
ON BAR 2 OF cust_pop DO cust_bar2
ON BAR 3 OF cust_pop DO cust_bar3
ON BAR 4 OF cust_pop DO cust_bar4
ON BAR 5 OF cust_pop DO cust_bar5
ON BAR 6 OF cust_pop DO cust_bar6
DO WHILE PROMPT()<>"Return to Main Menu"
   ACTIVATE POPUP cust_pop
ENDDO (PROMPT()<>"Return to Main Menu")
* ON SELECTION POPOP cust_pop DO cust_pro
*!*******************************************************************
*!
*!      Procedure: CUST_PRO
*!
*!      Called by: CUSTUNIT.PRG
*!
*!          Calls: UNITDATA            (procedure in CUSTUNIT.PRG)
*!               : CHANGELI            (procedure in EDITUNIT.PRG)
*!               : ADDLINCA            (procedure in EDITUNIT.PRG)
*!               : DELLIN              (procedure in EDITUNIT.PRG)
*!               : CHANGEEH            (procedure in EDITUNIT.PRG)
*!               : ENDUNIT             (procedure in EDITUNIT.PRG)
*!
*!*******************************************************************
PROCEDURE cust_pro
DO CASE
   * Select Unit
CASE BAR() = 1
   DO unitdata
```

131

```
          * Change LIN quantities
CASE BAR() = 2
    DO changeli
    * Add LIN
CASE BAR() = 3
    DO addlinca
    * Delete LIN
CASE BAR() = 4
    DO dellin
    * Change Unit Echelon
CASE BAR() = 5
    DO changeeh
    * Done
CASE BAR() = 6
    DO endunit
ENDCASE
*!********************************************************************
*!
*!          Procedure: STARTUNI
*!
*!          Called by: EDITUNIT.PRG
*!                   : CUSTUNIT.PRG
*!
*!              Calls: STARTHLP.PRG
*!                   : HELP.PRG
*!                   : ENDUNIT                    (procedure in EDITUNIT.PRG)
*!
*!               Uses: CSRCLIST.DBF
*!                   : LINLIST.DBF
*!                   : AMMORATE.DBF
*!                   : POLRATE.DBF
*!                   : SRCLIST.DBF
*!                   : SRCLINQT.DBF
*!                   : CSRCLINQ.DBF
*!                   : ECHELONS.DBF
*!
*!          MDX files: CSRCLIST.MDX
*!                   : LINLIST.MDX
*!                   : AMMORATE.MDX
*!                   : POLRATE.MDX
*!                   : SRCLIST.MDX
*!                   : SRCLINQT.MDX
*!                   : CSRCLINQ.MDX
*!                   : ECHELONS.MDX
*!
*!********************************************************************
PROCEDURE startuni
CLOSE DATA
CLEAR
DO starthlp
SELECT 8
USE ammorate ORDER lin
SELECT 9
USE polrate ORDER lin
SELECT 5
USE srclist NOUPDATE
SELECT 6
USE srclinqt ORDER lin NOUPDATE
SELECT 2
USE csrclist ORDER unitname
```

132

```
SELECT 14
USE csrclinq ORDER unitname
SELECT 25
USE echelons
SELECT 2
*!*********************************************************************
*!
*!       Procedure: UNITDATA
*!
*!       Called by: CUST_PRO               (procedure in CUSTUNIT.PRG)
*!
*!           Calls: ECHELON.PRG
*!                : DISPUNIT                (procedure in CREATESR.PRG)
*!
*!            Uses: TEMP.DBF
*!
*!*********************************************************************
PROCEDURE unitdata
SELECT 2
COPY STRUCTURE TO temp
SELECT 11
USE temp
APPEND BLANK
REPLACE src WITH "CUSTOM"
CLEAR
DO WHILE .T.
   *@ 0,2 SAY
"▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢▢"
   @ 2,1 SAY "SRC (Optional): "
   @ ROW(), COL() GET src
   @ 4,1 SAY "Unit Name: "
   @ 4,12 GET unitname PICTURE
"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
   @ 6,28 SAY "Authorized Strength: "
   @ 6,49 GET strength PICTURE "999999"
   @ 8,12 SAY "Organizational relationships and logistical caluculations"
   @ 9,12 SAY "are based on the Unit Name field.  You will NOT be allowed"
   @ 10,12 SAY "to change the Unit Name once you leave this screen, although"
   @ 11,12 SAY "you will be able to copy from a previously defined SRC."
   @ 17,31 SAY "CTRL-W when Done!"
   READ
   * Make sure a name was added
   IF unitname=" "
      RETURN
   ENDIF (unitname=" ")
   * Error check to make sure unitname is not a duplicate
   STORE temp->unitname TO fieldvar
   SELECT 2
   IF KEYMATCH(fieldvar,TAGNO("UNITNAME"))
      ? CHR(7)
      SET MESSAGE TO "You have duplicated a Unit Name already in use!"
      SELECT 11
      LOOP
   ENDIF (KEYMATCH(fieldvar,TAGNO("UNITNAME")))
   SELECT 11
   * Error check to make sure SRC is not a duplicate
   SELECT 11
   fieldvar=src
   IF src<>" "
      SELECT 5
```

133

```
            IF KEYMATCH(fieldvar,TAGNO("SRC"))
                ? CHR(7)
                SET MESSAGE TO "You have duplicated an SRC already in use!"
                SELECT 11
                LOOP
            ELSE
                SELECT 11
            ENDIF (KEYMATCH(fieldvar,TAGNO("SRC")))
        ENDIF (src<>" ")
        IF LASTKEY()=23
            EXIT
        ENDIF (LASTKEY()=23)
    ENDDO (.T.)
    SET MESSAGE TO
    STORE unitname TO fieldvar
    SELECT 11
    USE
    SELECT 2
    APPEND FROM temp
    DO echelon
    statusvar=2
    CLEAR
    DO dispunit
    * Change LIN quantities
    *!*****************************************************************
    *!
    *!        Procedure: CHANGELI
    *!
    *!        Called by: EDIU_PRO              (procedure in EDITUNIT.PRG)
    *!                 : CUST_PRO              (procedure in CUSTUNIT.PRG)
    *!
    *!           Calls: SETBROWS.PRG
    *!                 : ENDBROWS.PRG
    *!                 : DISPUNIT              (procedure in CREATESR.PRG)
    *!
    *!*****************************************************************
    PROCEDURE changeli
    IF statusvar=1
        ACTIVATE WINDOW small
        ? CHR(7)
        @ 1,2 SAY "You must create a unit first!"
        @ 2,2 SAY "Press any key..."
        WAIT ""
        DEACTIVATE WINDOW small
        RETURN
    ENDIF (statusvar=1)
    SELECT 14
    SET ORDER TO lin
    SET RELATION TO lin INTO ammorate, lin INTO polrate
    * limit csrclist access to records with the same unit name
    SET ORDER TO unitname
    SET KEY TO csrclist->unitname
    @ 0,25 SAY "POL-User"
    @ 0,55 SAY "AMMO-User"
    ACTIVATE WINDOW browser
    DO setbrows
    BROWSE FIELDS lin,quantity,polrate->lin_nomen /28,ammorate->lin_nomen FREEZE
    quantity
    DO endbrows
    DEACTIVATE WINDOW browser
```

```
@ 0,25 SAY "          "
@ 0,55 SAY "          "
SET MESSAGE TO
SET RELATION TO
SET KEY TO
DO dispunit
* Add LIN
*!*****************************************************************
*!
*!       Procedure: ADDLINCA
*!
*!       Called by: EDIU_PRO            (procedure in EDITUNIT.PRG)
*!              : CUST_PRO            (procedure in CUSTUNIT.PRG)
*!
*!          Calls: SETBROWS.PRG
*!              : ENDBROWS.PRG
*!              : DISPUNIT            (procedure in CREATESR.PRG)
*!
*!*****************************************************************
PROCEDURE addlinca
IF statusvar=1
   ACTIVATE WINDOW small
   ? CHR(7)
   @ 1,2 SAY "You must create a unit first!"
   @ 2,2 SAY "Press any key..."
   WAIT ""
   DEACTIVATE WINDOW small
   RETURN
ENDIF (statusvar=1)
SELECT 6
SET ORDER TO lin
SET RELATION TO lin INTO ammorate, lin INTO polrate
SET ORDER TO uniqueli
@ 0,25 SAY "POL-User"
@ 0,55 SAY "AMMO-User"
ACTIVATE WINDOW browser
DO setbrows
BROWSE FIELDS lin, polrate->lin_nomen, ammorate->lin_nomen
DO endbrows
DEACTIVATE WINDOW browser
@ 0,25 SAY "          "
@ 0,55 SAY "          "
STORE lin TO field1
STORE 0 TO field2
SET MESSAGE TO
ACTIVATE WINDOW editor
@ 2,2 SAY "Enter quantity of 0 if you do not want to add this LIN"
@ 4,10 SAY "LIN: "+field1
@ 5,10 SAY "Quantity: "
@ 5,21 GET field2
READ
SET RELATION TO
IF field2>0
   SELECT 14
   STORE src TO field3
   * Check to make sure LIN is not already included!
   SEEK field1
   IF .NOT. FOUND()
      APPEND BLANK
      REPLACE src WITH field3
```

135

```
        REPLACE lin WITH field1
        REPLACE quantity WITH field2
        REPLACE unitname WITH csrclist->unitname
    ENDIF (.NOT. FOUND())
    DEACTIVATE WINDOW editor
    SET KEY TO csrclist->unitname
 ENDIF (field2>0)
 DEACTIVATE WINDOW editor
 SELECT 2
 DO dispunit
 * Delete LIN
 *!********************************************************************
 *!
 *!        Procedure: DELLIN
 *!
 *!        Called by: EDIU_PRO          (procedure in EDITUNIT.PRG)
 *!                 : CUST_PRO          (procedure in CUSTUNIT.PRG)
 *!
 *!            Calls: SETBROWS.PRG
 *!                 : ENDBROWS.PRG
 *!                 : DISPUNIT          (procedure in CREATESR.PRG)
 *!
 *!********************************************************************
 PROCEDURE dellin
 IF statusvar=1
    ACTIVATE WINDOW small
    ? CHR(7)
    @ 1,2 SAY "You must choose a unit first!"
    @ 2,2 SAY "Press any key..."
    WAIT ""
    DEACTIVATE WINDOW small
    RETURN
 ENDIF (statusvar=1)
 SELECT 14
 SET ORDER TO lin
 SET RELATION TO lin INTO ammorate, lin INTO polrate
 * limit csrclist access to records with the same unit name
 SET ORDER TO unitname
 SET KEY TO cunit->unitname
 @ 0,25 SAY "POL-User"
 @ 0,55 SAY "AMMO-User"
 ACTIVATE WINDOW browser
 DO setbrows
 BROWSE FIELDS lin,quantity,polrate->lin_nomen /28,ammorate->lin_nomen FREEZE
 quantity NOEDIT
 DO endbrows
 DEACTIVATE WINDOW browser
 @ 0,25 SAY "             "
 @ 0,55 SAY "             "
 SET MESSAGE TO
 * Make sure user wants to delete LIN
 ACTIVATE WINDOW small
 STORE "   " TO MEMVAR
 @ 1,1 SAY "LIN: "+lin
 @ 2,1 SAY "POL-User "+polrate->lin_nomen
 @ 3,1 SAY "AMMO-User "+ammorate->lin_nomen
 @ 4,1 SAY "Type YES to Delete"
 @ 4,21 GET MEMVAR PICTURE "!!!"
 READ
 CLEAR
```

136

```
IF MEMVAR="YES"
    DELETE
    PACK
    WAIT "LIN DELETED... Press Any Key"
ELSE
    WAIT "LIN NOT DELETED... Press Any Key"
ENDIF (MEMVAR="YES")
DEACTIVATE WINDOW small
SET RELATION TO
SET KEY TO
DO dispunit
* Change Unit Echelon
*!******************************************************************
*!
*!        Procedure: CHANGEEH
*!
*!        Called by: EDIU_PRO          (procedure in EDITUNIT.PRG)
*!                 : CUST_PRO          (procedure in CUSTUNIT.PRG)
*!
*!            Calls: SETBROWS.PRG
*!                 : ENDBROWS.PRG
*!                 : DISPUNIT          (procedure in CREATESR.PRG)
*!
*!             Uses: ECHELONS.DBF
*!
*!        MDX files: ECHELONS.MDX
*!
*!******************************************************************
PROCEDURE changeeh
SELECT 25
USE echelons
DO setbrows
SET MESSAGE TO "No echelon estimated. Please choose unit echelon, then Enter to
Select."
ACTIVATE WINDOW browser
BROWSE NOEDIT NOMENU NOAPPEND NODELETE FIELDS echelon
DO endbrows
DEACTIVATE WINDOW browser
SELECT 2
REPLACE echelon WITH echelons->echelon
DO dispunit
* Return to previous menu
*!******************************************************************
*!
*!        Procedure: ENDUNIT
*!
*!        Called by: STARTUNI          (procedure in EDITUNIT.PRG)
*!                 : EDIU_PRO          (procedure in EDITUNIT.PRG)
*!                 : CUST_PRO          (procedure in CUSTUNIT.PRG)
*!
*!******************************************************************
PROCEDURE endunit
CLEAR
DEACTIVATE POPUP
CLOSE DATABASES
RETURN
* Display record information
*!******************************************************************
*!
*!        Procedure: DISPUNIT
```

```
*!
*!        Called by: COPYSRC              (procedure in CREATESR.PRG)
*!                 : COPYCSRC             (procedure in CREATESR.PRG)
*!                 : ADDDATA              (procedure in CREATESR.PRG)
*!                 : SELECTUNI            (procedure in EDITUNIT.PRG)
*!                 : SELECTUN2            (procedure in EDITUNIT.PRG)
*!                 : CHANGELI             (procedure in EDITUNIT.PRG)
*!                 : ADDLINCA             (procedure in EDITUNIT.PRG)
*!                 : DELLIN               (procedure in EDITUNIT.PRG)
*!                 : CHANGEEH             (procedure in EDITUNIT.PRG)
*!                 : EDITSTR              (procedure in EDITUNIT.PRG)
*!                 : UNITDATA             (procedure in CUSTUNIT.PRG)
*!
*!**********************************************************************
PROCEDURE dispunit
SELECT 2
@ 16,1 TO 16,78
@ 18,5 SAY "SRC:       "+csrclist->src
@ 20,5 SAY "Unit:      "+unitname
@ 21,5 SAY "Echelon: "+echelon
IF echelon="COMPANY"
    @ ROW(), COL() SAY " Level or Below"
ELSE
    @ ROW(), COL() SAY "                    "
ENDIF (echelon="COMPANY")
@ 22,5 SAY "Strength "+TRANSFORM(strength,"999,999")
*: EOF: CUSTUNIT.PRG
```

```
*:********************************************************************
*:
*: Procedure file: D:\SURE\DELORDE.PRG
*:
*:   Procs & Fncts: DORD_PRO
*:
*:          Set by: ORDE_PRO              (procedure in SURE.PRG)
*:
*:           Calls: STARTHLP.PRG
*:                : DORD_PRO              (procedure in DELORDE.PRG)
*:
*:            Uses: ORDERTAS.DBF
*:                : ORDERLST.DBF
*:
*:       MDX files: ORDERTAS.MDX
*:                : ORDERLST.MDX
*:
*:       Documented 04/16/94 at 11:01            SNAP!  version 5.00
*:********************************************************************
CLEAR
SET PROCEDURE TO instorde
STORE PROGRAM() TO progvar
STORE 1 TO statusvar
* Open Files
DO starthlp
STORE 1 TO statusvar
SELECT 18
USE ordertas
SELECT 3
USE orderlst
* Main Popup
DEFINE POPUP dord_pop FROM 1,23
DEFINE BAR 1 OF dord_pop PROMPT "Get Order"
DEFINE BAR 2 OF dord_pop PROMPT "Delete Order"
DEFINE BAR 3 OF dord_pop PROMPT "Return to Main Menu"
ON SELECTION POPUP dord_pop DO dord_pro
ON BAR 1 OF dord_pop DO dord_bar1
ON BAR 2 OF dord_pop DO dord_bar2
ON BAR 3 OF dord_pop DO dord_bar3
DO WHILE PROMPT()<>"Return to Main Menu"
   ACTIVATE POPUP dord_pop
ENDDO (PROMPT()<>"Return to Main Menu")
*!********************************************************************
*!
*!      Procedure: DORD_PRO
*!
*!      Called by: DELORDE.PRG
*!
*!          Calls: GETORDER              (procedure in EDITORDE.PRG)
*!
*!********************************************************************
PROCEDURE dord_pro
DO CASE
CASE BAR()=1
   DO getorder
   statusvar=2
CASE BAR()=2
   IF statusvar=1
      ACTIVATE WINDOW small
      ? CHR(7)
```

139

```
            @ 1,2 SAY "You must select an Order first!"
            @ 3,3 SAY "Press any key..."
            WAIT ""
            DEACTIVATE WINDOW small
            CLEAR
        ELSE
            STORE "NO " TO yesnovar
            ACTIVATE WINDOW small
            @ 1,1 SAY "Enter YES to Delete this Order! "
            @ ROW(), COL() GET yesnovar PICTURE "!!!"
            READ
            IF yesnovar="YES"
                STORE NUMBER TO numberva
                SELECT 18
                DELETE FOR NUMBER=numberva
                PACK
                SELECT 3
                DELETE FOR NUMBER=numberva
                PACK
                CLEAR
                @ 2,5 SAY "Deleted! Press any key..."
                WAIT ""
                DEACTIVATE WINDOW small
                CLEAR
            ELSE
                @ 2,5 SAY "NOT Deleted! Press any key..."
                WAIT ""
            ENDIF (yesnovar="YES")
        ENDIF (statusvar=1)
        DEACTIVATE WINDOW small
    CASE BAR()=3
        CLOSE DATA
        DEACTIVATE POPUP
        RETURN
    ENDCASE
* CASE BAR()=1
*CLEAR
*!*************************************************************
*!
*!      Procedure: GETORDER
*!
*!      Called by: ORDT_PRO              (procedure in EDITORDE.PRG)
*!               : DORD_PRO              (procedure in DELORDE.PRG)
*!               : ORDR_PRO              (procedure in ORDEREP1.PRG)
*!
*!         Calls: SETBROWS.PRG
*!              : ENDBROWS.PRG
*!              : DISPORD2               (procedure in CREATEOR.PRG)
*!              : VIEWORDE               (procedure in CREATEOR.PRG)
*!              : DISPORD                (procedure in CREATEOR.PRG)
*!              : DISPORD3               (procedure in ORDEREP1.PRG)
*!              : CALCPOL.PRG
*!              : CALCAMMO.PRG
*!              : CALCOTHE.PRG
*!
*!*************************************************************
PROCEDURE getorder
DO vieworde
DO dispord
statusvar=2
```

140

```
*!********************************************************************
*!
*!          Procedure: DISPORD
*!
*!          Called by: CREATEO              (procedure in CREATEOR.PRG)
*!                   : GETORDER             (procedure in EDITORDE.PRG)
*!
*!********************************************************************
PROCEDURE dispord
@ 17,2 TO 17,78
@ 19,2 SAY "Order Number:   "
@ ROW(),COL() SAY NUMBER
@ 21,2 SAY "  Order Name:   "
@ ROW(),COL() SAY name
*!********************************************************************
*!
*!          Procedure: DISPORD2
*!
*!          Called by: VIEWORDE             (procedure in CREATEOR.PRG)
*!                   : ORDT_PRO             (procedure in EDITORDE.PRG)
*!                   : GETORDER             (procedure in EDITORDE.PRG)
*!
*!********************************************************************
PROCEDURE dispord2
@ 1,2 SAY "Order Number:   "
@ ROW(),COL() SAY NUMBER
@ 2,2 SAY "  Order Name: - "
@ ROW(),COL() SAY name
@ 3,2 SAY "Description:   "
@ ROW(), COL() SAY descriptio
@ 4,2 SAY "Number of Phases:   "
@ ROW(), COL() SAY LTRIM(STR(phases))
* CASE BAR()=2
*!********************************************************************
*!
*!          Procedure: VIEWORDE
*!
*!          Called by: ORDS_PRO             (procedure in CREATEOR.PRG)
*!                   : ORDT_PRO             (procedure in EDITORDE.PRG)
*!                   : GETORDER             (procedure in EDITORDE.PRG)
*!
*!              Calls: SETBROWS.PRG
*!                   : ENDBROWS.PRG
*!                   : DISPORD2             (procedure in CREATEOR.PRG)
*!                   : DISPORD3             (procedure in ORDEREP1.PRG)
*!
*!********************************************************************
PROCEDURE vieworde
SELECT 3
* Make sure there are some records to view
IF RECCOUNT()=0
   ACTIVATE WINDOW small
   ? CHR(7)
   @ 1,2 SAY "You must have at"
   @ 2,2 SAY "least one Order!"
   @ 3,3 SAY "Press any key..."
   WAIT ""
   DEACTIVATE WINDOW small
   CLEAR
   RETURN
```

```
ENDIF (RECCOUNT()=0)
ACTIVATE WINDOW browser
GO TOP
*      SET MESSAGE TO "Choose an Order to View"
DO setbrows
BROWSE NOMENU NOEDIT NOAPPEND NODELETE FIELDS NUMBER,name, descriptio FREEZE
name COMPRESS
DO endbrows
DEACTIVATE WINDOW browser
SET CURSOR OFF
CLEAR
DO WHILE LASTKEY()<>23
    DO dispord2
    IF RECCOUNT()=1
       @ 23,30 SAY "Crtl-W when Done"
    ELSE
       @ 23,15 SAY "Page-Up or Page-Down to View, Ctrl-W when Done"
    ENDIF (RECCOUNT()=1)
    IF LASTKEY()=3
       SKIP
       IF EOF()
          GO TOP
       ENDIF (EOF())
    ENDIF (LASTKEY()=3)
    IF LASTKEY()=18
       SKIP -1
       IF BOF()
          GO BOTTOM
       ENDIF (BOF())
    ENDIF (LASTKEY()=18)
    WAIT ""
ENDDO (LASTKEY()<>23)
SET CURSOR ON
SET MESSAGE TO
RETURN
*: EOF: DELORDE.PRG
```

```
*:********************************************************************.*
*:
*: Procedure file: D:\SURE\DELPROF.PRG
*:
*:   Procs & Fncts: DELP_PRO
*:                : ENDDELP
*:                : DELPRO
*:                : SELECTPRO
*:
*:          Set by: PROF_PRO              (procedure in SURE.PRG)
*:
*:           Calls: STARTED               (procedure in DELUNIT.PRG)
*:                : DELP_PRO              (procedure in DELPROF.PRG)
*:
*:          Documented 04/16/94 at 11:01                SNAP!  version 5.00
*:********************************************************************
*delprof.prg
STORE PROGRAM() TO progvar
* PRIVATE Variables
CLEAR
STORE 0 TO countvar
STORE SPACE(60) TO namevar
STORE SPACE(60) TO profilevar
STORE 1 TO statusvar
STORE "NO " TO yesnovar
SET PROCEDURE TO instprof
DO started
IF BOF()
   ACTIVATE WINDOW small
   ? CHR(7)
   @ 1,2 SAY "No Profiles Exist"
   @ 2,2 SAY "Press any key..."
   WAIT ""
   DEACTIVATE WINDOW small
   RETURN
ENDIF (BOF())
SET MESSAGE TO "Select Parameter Set to Delete"
DEFINE POPUP delp_pop FROM 1,5
DEFINE BAR 1 OF delp_pop PROMPT "Select Parameter Set"
DEFINE BAR 2 OF delp_pop PROMPT "Delete Parameter Set"
DEFINE BAR 3 OF delp_pop PROMPT "Return to Main Menu"
ON SELECTION POPUP delp_pop DO delp_pro
ON BAR 1 OF delp_pop DO delp_bar1
ON BAR 2 OF delp_pop DO delp_bar2
ON BAR 3 OF delp_pop DO delp_bar3
DO WHILE PROMPT()<>"Return to Main Menu"
   ACTIVATE POPUP delp_pop
ENDDO (PROMPT()<>"Return to Main Menu")
*!********************************************************************
*!
*!      Procedure: DELP_PRO
*!
*!      Called by: DELPROF.PRG
*!
*!          Calls: SELECTPRO              (procedure in DELPROF.PRG)
*!               : DELPRO                 (procedure in DELPROF.PRG)
*!               : ENDDELP                (procedure in DELPROF.PRG)
*!
*!********************************************************************
PROCEDURE delp_pro
```

143

```
DO CASE
CASE BAR()=1
   DO selectpro
CASE BAR()=2
   DO delpro
CASE BAR()=3
   DO enddelp
ENDCASE
* BAR()=3
*!******************************************************************
*!
*!        Procedure: ENDDELP
*!
*!        Called by: DELP_PRO            (procedure in DELPROF.PRG)
*!
*!******************************************************************
PROCEDURE enddelp
CLOSE DATA
DEACTIVATE POPUP
CLEAR
RETURN
*!******************************************************************
*!
*!        Procedure: DELPRO
*!
*!        Called by: DELP_PRO            (procedure in DELPROF.PRG)
*!
*!******************************************************************
PROCEDURE delpro
IF statusvar=1
   ACTIVATE WINDOW small
   ? CHR(7)
   @ 1,2 SAY "Select a Parameter Set first!"
   @ 3,3 SAY "Press any key..."
   WAIT ""
   DEACTIVATE WINDOW small
   CLEAR
ELSE
   STORE "NO " TO yesnovar
   ACTIVATE WINDOW small
   @ 1,1 SAY "Enter YES to Delete Parameter Set "
   @ ROW(), COL() GET yesnovar PICTURE "!!!"
   READ
   IF yesnovar="YES"
      STORE cprofile->profilenam TO nameva
      * check to make sure this Profile is not used in any task organizations!
      SELECT 18
      SET ORDER TO profilenam
      GO TOP
      SEEK nameva
      IF FOUND()
         DEACTIVATE WINDOW small
         ACTIVATE WINDOW process
         ? CHR(7)
         CLEAR
         @ 1,1 SAY "Cannot Delete! Parameter Set is"
         @ 2,1 SAY "used by Order number:   "+ordertas->number
         @ 3,1 SAY "You must remove it from the order"
         @ 4,1 SAY "before it can be deleted."
         WAIT
```

144

```
                DEACTIVATE WINDOW process     ·
                SELECT 4
            ELSE
                SELECT 4
                GO TOP
                DELETE FOR profilenam=nameva
                PACK
                CLEAR
                @ 2,5 SAY "Deleted! Press any key..."
                WAIT ""
                DEACTIVATE WINDOW small
                CLEAR
                SELECT 4
            ENDIF (FOUND())
        ELSE
            CLEAR
            @ 2,5 SAY "NOT Deleted! Press any key..."
            WAIT ""
        ENDIF (yesnovar="YES")
    ENDIF (statusvar=1)
    DEACTIVATE WINDOW small
*!***********************************************************************
*!
*!        Procedure: STARTED
*!
*!        Called by: DELUNIT.PRG
*!                 : EDITTASK.PRG
*!                 : DELTASK.PRG
*!                 : DELPROF.PRG
*!
*!            Calls: STARTHLP.PRG
*!
*!             Uses: CSRCLIST.DBF
*!                 : CSRCLINQ.DBF
*!                 : TASKORG.DBF
*!                 : ORDERTAS.DBF
*!                 : CPROFILE.DBF
*!
*!        MDX files: CSRCLIST.MDX
*!                 : CSRCLINQ.MDX
*!                 : TASKORG.MDX
*!                 : ORDERTAS.MDX
*!                 : CPROFILE.MDX
*!
*!***********************************************************************
PROCEDURE started
CLEAR
DO starthlp
SELECT 18
USE ordertas
SET ORDER TO profilenam
SELECT 4
USE cprofile
SET ORDER TO profilenam
*!***********************************************************************
*!
*!        Procedure: SELECTPRO
*!
*!        Called by: DELP_PRO              (procedure in DELPROF.PRG)
*!
```

145

```
*!          Calls: SETBROWS.PRG
*!               : ENDBROWS.PRG
*!               : DISPPROF           (procedure in CREATEPR.PRG)
*!
*!***********************************************************************
PROCEDURE selectpro
SELECT 4
SET ORDER TO profilenam
GO TOP
ACTIVATE WINDOW browser
DO setbrows
BROWSE NOMENU NOEDIT NODELETE NOAPPEND FIELDS profilenam
DO endbrows
STORE profilenam TO fieldvar
DEACTIVATE WINDOW browser
SET MESSAGE TO
statusvar=2
DO dispprof
* Display record information
*!***********************************************************************
*!
*!        Procedure: DISPPROF
*!
*!        Called by: NEWPROFI           (procedure in CREATEPR.PRG)
*!                 : VIEWPROF           (procedure in CREATEPR.PRG)
*!                 : SELEPROF           (procedure in EDITPROF.PRG)
*!                 : CHANGEAP           (procedure in EDITPROF.PRG)
*!                 : CHANGEAC           (procedure in EDITPROF.PRG)
*!                 : CHANGEI            (procedure in EDITPROF.PRG)
*!                 : CHANGEIF           (procedure in EDITPROF.PRG)
*!                 : CHANGECI           (procedure in EDITPROF.PRG)
*!                 : CHANGEH2           (procedure in EDITPROF.PRG)
*!                 : CHANGEME           (procedure in EDITPROF.PRG)
*!                 : SELECTPRO          (procedure in DELPROF.PRG)
*!                 : FINDPRO            (procedure in CREATEOR.PRG)
*!
*!***********************************************************************
PROCEDURE dispprof
SELECT 2
@ 16,1 TO 16,78
@ 18,5 SAY "Profile: "+cprofile->profilenam

*: EOF: DELPROF.PRG
```

```
*:********************************************************************
*:
*:  Procedure file: D:\SURE\DELTASK.PRG
*:
*:   Procs & Fncts: DELT_PRO
*:               : ENDDELT
*:               : DELTAS
*:
*:          Set by: TASK_PRO              (procedure in SURE.PRG)
*:
*:          Calls: STARTED                (procedure in DELUNIT.PRG)
*:               : DELT_PRO               (procedure in DELTASK.PRG)
*:
*:       Documented 04/16/94 at 11:01               SNAP!  version 5.00
*:********************************************************************
*edittask.prg
STORE PROGRAM() TO progvar
* PRIVATE Variables
CLEAR
STORE 0 TO countvar
STORE SPACE(60) TO namevar
STORE SPACE(60) TO unitvar
STORE 1 TO statusvar
STORE "NO " TO yesnovar
SET PROCEDURE TO insttask
DO started
IF BOF()
   ACTIVATE WINDOW small
   ? CHR(7)
   @ 1,2 SAY "No Task Organizations Exist"
   @ 2,2 SAY "Press Return to continue..."
   WAIT ""
   DEACTIVATE WINDOW small
   RETURN
ENDIF (BOF())
SET MESSAGE TO "Select Task Force to Delete"
DEFINE POPUP delt_pop FROM 1,5
DEFINE BAR 1 OF delt_pop PROMPT "Select Task Organization"
DEFINE BAR 2 OF delt_pop PROMPT "Delete Task Organization"
DEFINE BAR 3 OF delt_pop PROMPT "Return to Main Menu"
ON SELECTION POPUP delt_pop DO delt_pro
ON BAR 1 OF delt_pop DO delt_bar1
ON BAR 2 OF delt_pop DO delt_bar2
ON BAR 3 OF delt_pop DO delt_bar3
DO WHILE PROMPT()<>"Return to Main Menu"
   ACTIVATE POPUP delt_pop
ENDDO (PROMPT()<>"Return to Main Menu")
*!********************************************************************
*!
*!      Procedure: DELT_PRO
*!
*!      Called by: DELTASK.PRG
*!
*!          Calls: FINDTA                 (procedure in CREATETC.PRG)
*!               : DELTAS                 (procedure in DELTASK.PRG)
*!               : ENDDELT                (procedure in DELTASK.PRG)
*!
*!********************************************************************
PROCEDURE delt_pro
DO CASE
```

147

```
       CASE BAR()=1
          DO findta
       CASE BAR()=2
          DO deltas
       CASE BAR()=3
          DO enddelt
       ENDCASE
       *  BAR()=1
       *  Select Task Organization
       *!********************************************************************
       *!
       *!         Procedure: FINDTA
       *!
       *!         Called by: EDITTASK.PRG
       *!                  : COPT_PRO              (procedure in CREATETC.PRG)
       *!                  : EDIT_PRO              (procedure in EDITTASK.PRG)
       *!                  : DELT_PRO              (procedure in DELTASK.PRG)
       *!
       *!             Calls: SETBROWS.PRG
       *!                  : ENDBROWS.PRG
       *!                  : DISPTASK              (procedure in CREATETA.PRG)
       *!
       *!********************************************************************
       PROCEDURE findta
       SELECT 1
       SET ORDER TO uniqueta                && only view each task order name once
       GO TOP
       ACTIVATE WINDOW browser
       DO setbrows
       BROWSE FIELDS taskorgnam NOMENU NOEDIT NOAPPEND NODELETE COMPRESS
       DO endbrows
       DEACTIVATE WINDOW browser
       STORE taskorgnam TO namevar
       DO disptask
       SET ORDER TO taskorgnam
       statusvar=2
       *  BAR()=3
       *!********************************************************************
       *!
       *!         Procedure: ENDDELT
       *!
       *!         Called by: DELT_PRO              (procedure in DELTASK.PRG)
       *!
       *!********************************************************************
       PROCEDURE enddelt
       CLOSE DATA
       DEACTIVATE POPUP
       CLEAR
       RETURN
       *!********************************************************************
       *!
       *!         Procedure: DELTAS
       *!
       *!         Called by: DELT_PRO              (procedure in DELTASK.PRG)
       *!
       *!********************************************************************
       PROCEDURE deltas
       IF statusvar=1
          ACTIVATE WINDOW small
          ? CHR(7)
```

148

```
       @ 1,2 SAY "You must select an Task Organization first!"
       @ 3,3 SAY "Press any key..."
       WAIT ""
       DEACTIVATE WINDOW small
       CLEAR
   ELSE
       STORE "NO " TO yesnovar
       ACTIVATE WINDOW small
       @ 1,1 SAY "Enter YES to Delete this Task"
       @ 2,1 SAY "Organization "
       @ ROW(), COL() GET yesnovar PICTURE "!!!"
       READ
       IF yesnovar="YES"
           STORE taskorgnam TO nameva
           * check to make sure this task organzation is not used in any orders!
           SELECT 18
           SET ORDER TO taskorgnam
           GO TOP
           SEEK nameva
           IF FOUND()
               ? CHR(7)
               CLEAR
               @ 1,1 SAY "Cannot Delete! Task Organization is"
               @ 2,1 SAY "used by Order Number "+ordertas->number
               @ 3,1 SAY "You must remove it from the order"
               @ 4,1 SAY "before it can be deleted."
               WAIT
               SELECT 1
           ELSE
               SELECT 1
               GO TOP
               DELETE FOR taskorgnam=nameva
               PACK
               CLEAR
               @ 2,5 SAY "Deleted! Press any key..."
               WAIT ""
               DEACTIVATE WINDOW small
               CLEAR
           ENDIF (FOUND())
       ELSE
           CLEAR
           @ 2,5 SAY "NOT Deleted! Press any key..."
           WAIT ""
       ENDIF (yesnovar="YES")
   ENDIF (statusvar=1)
   DEACTIVATE WINDOW small
*!*************************************************************************
*!
*!          Procedure: STARTED
*!
*!          Called by: DELUNIT.PRG
*!                    : EDITTASK.PRG
*!                    : DELTASK.PRG
*!                    : DELPROF.PRG
*!
*!              Calls: STARTHLP.PRG
*!
*!               Uses: CSRCLIST.DBF
*!                    : CSRCLINQ.DBF
*!                    : TASKORG.DBF
```

149

```
*!                   : ORDERTAS.DBF
*!                   : CPROFILE.DBF
*!
*!       MDX files: CSRCLIST.MDX
*!                 : CSRCLINQ.MDX
*!                 : TASKORG.MDX
*!                 : ORDERTAS.MDX
*!                 : CPROFILE.MDX
*!
*!*****************************************************************
PROCEDURE started
DO starthlp
SELECT 18
USE ordertas
CLEAR
SELECT 2
USE csrclist
SELECT 1
USE taskorg
SET ORDER TO taskorgnam
*!*****************************************************************
*!
*!       Procedure: DISPTASK
*!
*!       Called by: NAMETA              (procedure in CREATETA.PRG)
*!                 : ASSIGNUN           (procedure in CREATETA.PRG)
*!                 : ASSIGNU2           (procedure in CREATETA.PRG)
*!                 : FINDTA             (procedure in CREATETC.PRG)
*!                 : ASSIGNNA           (procedure in CREATETC.PRG)
*!                 : VIEWTASK           (procedure in EDITTASK.PRG)
*!                 : VIEWTAS2           (procedure in EDITTASK.PRG)
*!                 : UNASSIGU           (procedure in EDITTASK.PRG)
*!                 : UNASSIG2           (procedure in EDITTASK.PRG)
*!
*!*****************************************************************
PROCEDURE disptask
@ 17,2 TO 17,78
@ 21,2 SAY "Task Organization:   "
@ ROW(),COL() SAY namevar

*: EOF: DELTASK.PRG
```

```
*:*********************************************************************
*:
*: Procedure file: D:\SURE\DELUNIT.PRG
*:
*:  Procs & Fncts: STARTED
*:               : DELU_PRO
*:               : ENDDELU
*:               : DELUAS
*:
*:         Set by: SRC_PRO            (procedure in SURE.PRG)
*:
*:          Calls: STARTED            (procedure in DELUNIT.PRG)
*:               : DELU_PRO           (procedure in DELUNIT.PRG)
*:
*:         Documented 04/16/94 at 11:01              SNAP!  version 5.00
*:*********************************************************************
*edittask.prg
STORE PROGRAM() TO progvar
* PRIVATE Variables
CLEAR
STORE 0 TO countvar
STORE SPACE(60) TO namevar
STORE SPACE(60) TO unitvar
STORE 1 TO statusvar
STORE "NO " TO yesnovar
SET PROCEDURE TO instunit
DO started
IF BOF()
   ACTIVATE WINDOW small
   ? CHR(7)
   @ 1,2 SAY "No Units Exist"
   @ 2,2 SAY "Press any key..."
   WAIT ""
   DEACTIVATE WINDOW small
   CLOSE DATA
   RETURN
ENDIF (BOF())
SET MESSAGE TO "Select Unit to Delete"
DEFINE POPUP delu_pop FROM 1,5
DEFINE BAR 1 OF delu_pop PROMPT "Select Unit - Unit Name Order"
DEFINE BAR 2 OF delu_pop PROMPT "Select Unit - SRC Order"
DEFINE BAR 3 OF delu_pop PROMPT "Delete Unit"
DEFINE BAR 4 OF delu_pop PROMPT "Return to Main Menu"
ON SELECTION POPUP delu_pop DO delu_pro
ON BAR 1 OF delu_pop DO delu_bar1
ON BAR 2 OF delu_pop DO delu_bar2
ON BAR 3 OF delu_pop DO delu_bar3
ON BAR 4 OF delu_pop DO delu_bar4
DO WHILE PROMPT()<>"Return to Main Menu"
   ACTIVATE POPUP delu_pop
ENDDO (PROMPT()<>"Return to Main Menu")
*!*********************************************************************
*!
*!      Procedure: DELU_PRO
*!
*!      Called by: DELUNIT.PRG
*!
*!          Calls: SELECTUNI          (procedure in EDITUNIT.PRG)
*!               : SELECTUN2          (procedure in EDITUNIT.PRG)
*!               : DELUAS             (procedure in DELUNIT.PRG)
```

151

```
*!                       : ENDDELU                (procedure in DELUNIT.PRG)
*!
*!*************************************************************************
PROCEDURE delu_pro
DO CASE
CASE BAR()=1
   DO selectuni
CASE BAR()=2
   DO selectun2
CASE BAR()=3
   DO deluas
CASE BAR()=4
   DO enddelu
ENDCASE
*  BAR()=3
*!*************************************************************************
*!
*!        Procedure: ENDDELU
*!            .
*!        Called by: DELU_PRO            (procedure in DELUNIT.PRG)
*!
*!*************************************************************************
PROCEDURE enddelu
CLOSE DATA
DEACTIVATE POPUP
CLEAR
RETURN
*!*************************************************************************
*!
*!        Procedure: DELUAS
*!
*!        Called by: DELU_PRO            (procedure in DELUNIT.PRG)
*!
*!*************************************************************************
PROCEDURE deluas
IF statusvar=1
   ACTIVATE WINDOW small
   ? CHR(7)
   @ 1,2 SAY "You must select an Unit first!"
   @ 3,3 SAY "Press any key..."
   WAIT ""
   DEACTIVATE WINDOW small
   CLEAR
ELSE
   STORE "NO " TO yesnovar
   ACTIVATE WINDOW small
   @ 1,1 SAY "Enter YES to Delete this Unit "
   @ ROW(), COL() GET yesnovar PICTURE "!!!"
   READ
   IF yesnovar="YES"
      STORE unitname TO nameva
      * check to make sure this Unit is not used in any task organizations!
      SELECT 1
      SET ORDER TO unitname
      GO TOP
      SEEK nameva
      IF FOUND()
         DEACTIVATE WINDOW small
         ACTIVATE WINDOW process
         ? CHR(7)
```

152

```
                CLEAR
                @ 1,1 SAY "Cannot Delete! Unit is"
                @ 2,1 SAY "used by Task Organization:   "+taskorg->taskorgnam
                @ 3,1 SAY "You must remove it from the order"
                @ 4,1 SAY "before it can be deleted."
                WAIT
                DEACTIVATE WINDOW process
                SELECT 2
            ELSE
                SELECT 2
                GO TOP
                DELETE FOR unitname=nameva
                PACK
                SELECT 14
                GO TOP
                DELETE FOR unitname=nameva
                PACK
                CLEAR
                @ 2,5 SAY "Deleted! Press any key..."
                WAIT ""
                DEACTIVATE WINDOW small
                CLEAR
                SELECT 2
            ENDIF (FOUND())
        ELSE
            CLEAR
            @ 2,5 SAY "NOT Deleted! Press any key..."
            WAIT ""
        ENDIF (yesnovar="YES")
    ENDIF (statusvar=1)
    DEACTIVATE WINDOW small
    *!*******************************************************************
    *!
    *!       Procedure: STARTED
    *!
    *!       Called by: DELUNIT.PRG
    *!                 : EDITTASK.PRG
    *!                 : DELTASK.PRG
    *!                 : DELPROF.PRG
    *!
    *!           Calls: STARTHLP.PRG
    *!
    *!            Uses: CSRCLIST.DBF
    *!                : CSRCLINQ.DBF
    *!                : TASKORG.DBF
    *!                : ORDERTAS.DBF
    *!                : CPROFILE.DBF
    *!
    *!       MDX files: CSRCLIST.MDX
    *!                : CSRCLINQ.MDX
    *!                : TASKORG.MDX
    *!                : ORDERTAS.MDX
    *!                : CPROFILE.MDX
    *!
    *!*******************************************************************
    PROCEDURE started
    CLEAR
    DO starthlp
    SELECT 2
    USE csrclist
```

```
SET ORDER TO unitname
SELECT 14
USE csrclinq
SELECT 1
USE taskorg
SET ORDER TO unitname
SELECT 2
*!********************************************************************
*!
*!        Procedure: SELECTUNI
*!
*!        Called by: EDIU_PRO          (procedure in EDITUNIT.PRG)
*!                 : DELU_PRO          (procedure in DELUNIT.PRG)
*!
*!            Calls: SETBROWS.PRG
*!                 : ENDBROWS.PRG
*!                 : DISPUNIT          (procedure in CREATESR.PRG)
*!
*!********************************************************************
PROCEDURE selectuni
SELECT 2
SET ORDER TO unitname
GO TOP
ACTIVATE WINDOW browser
DO setbrows
BROWSE NOMENU NOEDIT NODELETE NOAPPEND FIELDS src,unitname
DO endbrows
STORE unitname TO fieldvar
DEACTIVATE WINDOW browser
SET MESSAGE TO
statusvar=2
DO dispunit
*!********************************************************************
*!
*!        Procedure: SELECTUN2
*!
*!        Called by: EDIU_PRO          (procedure in EDITUNIT.PRG)
*!                 : DELU_PRO          (procedure in DELUNIT.PRG)
*!
*!            Calls: SETBROWS.PRG
*!                 : ENDBROWS.PRG
*!                 : DISPUNIT          (procedure in CREATESR.PRG)
*!
*!********************************************************************
PROCEDURE selectun2
SELECT 2
SET ORDER TO unitname
GO TOP
ACTIVATE WINDOW browser
DO setbrows
SET ORDER TO src
BROWSE NOMENU NOEDIT NODELETE NOAPPEND FIELDS src,unitname
SET ORDER TO unitname
DO endbrows
STORE unitname TO fieldvar
DEACTIVATE WINDOW browser
SET MESSAGE TO
statusvar=2
DO dispunit
* Display record information
```

154

```
*!******************************************************************
*!                                     ,
*!        Procedure: DISPUNIT
*!
*!        Called by: COPYSRC              (procedure in CREATESR.PRG)
*!                 : COPYCSRC             (procedure in CREATESR.PRG)
*!                 : ADDDATA              (procedure in CREATESR.PRG)
*!                 : SELECTUNI            (procedure in EDITUNIT.PRG)
*!                 : SELECTUN2            (procedure in EDITUNIT.PRG)
*!                 : CHANGELI             (procedure in EDITUNIT.PRG)
*!                 : ADDLINCA             (procedure in EDITUNIT.PRG)
*!                 : DELLIN               (procedure in EDITUNIT.PRG)
*!                 : CHANGEEH             (procedure in EDITUNIT.PRG)
*!                 : EDITSTR              (procedure in EDITUNIT.PRG)
*!                 : UNITDATA             (procedure in CUSTUNIT.PRG)
*!
*!******************************************************************
PROCEDURE dispunit
SELECT 2
@ 16,1 TO 16,78
@ 18,5 SAY "Unit: "+unitname

*: EOF: DELUNIT.PRG
```

```
*:*********************************************************************
*:                              '
*:        Program: D:\SURE\ECHELON.PRG
*:
*:      Called by: ADDDATA              (procedure in CREATESR.PRG)
*:              : UNITDATA              (procedure in CUSTUNIT.PRG)
*:
*:        Documented 04/16/94 at 11:02                SNAP!  version 5.00
*:*********************************************************************
* Echelon.prg  Attempts to assign proper echelon to unit
DO CASE
CASE RAT(" CO ",UPPER(unitname))<>0
   REPLACE echelon WITH "COMPANY"
   RETURN
CASE RAT(" CO",UPPER(unitname))<>0
   REPLACE echelon WITH "COMPANY"
   RETURN
CASE RAT("COMPANY",UPPER(unitname))<>0
   REPLACE echelon WITH "COMPANY"
   RETURN
CASE RAT("Troop",UPPER(unitname))<>0
   REPLACE echelon WITH "COMPANY"
   RETURN
CASE RAT("TRP",UPPER(unitname))<>0
   REPLACE echelon WITH "COMPANY"
   RETURN
CASE RAT("BATTERY",UPPER(unitname))<>0
   REPLACE echelon WITH "COMPANY"
   RETURN
CASE RAT("BTRY",UPPER(unitname))<>0
   REPLACE echelon WITH "COMPANY"
   RETURN
CASE RAT("HHT",UPPER(unitname))<>0
   REPLACE echelon WITH "COMPANY"
   RETURN
CASE RAT("HHSC",UPPER(unitname))<>0
   REPLACE echelon WITH "COMPANY"
   RETURN
CASE RAT("HHS",UPPER(unitname))<>0
   REPLACE echelon WITH "COMPANY"
   RETURN
CASE RAT("HHD",UPPER(unitname))<>0
   REPLACE echelon WITH "COMPANY"
   RETURN
CASE RAT("HB",UPPER(unitname))<>0
   REPLACE echelon WITH "COMPANY"
   RETURN
CASE RAT("HHC",UPPER(unitname))<>0
   REPLACE echelon WITH "COMPANY"
   RETURN
CASE RAT("BAND",UPPER(unitname))<>0
   REPLACE echelon WITH "COMPANY"
   RETURN
CASE RAT("ELEMENT",UPPER(unitname))<>0
   REPLACE echelon WITH "COMPANY"
   RETURN
CASE RAT("SECTION",UPPER(unitname))<>0
   REPLACE echelon WITH "COMPANY"
   RETURN
CASE RAT("SEC ",UPPER(unitname))<>0
```

```
      REPLACE echelon WITH "COMPANY"
      RETURN
CASE RAT("SEC,",UPPER(unitname))<>0
      REPLACE echelon WITH "COMPANY"
      RETURN
CASE RAT("PLATOON",UPPER(unitname))<>0
      REPLACE echelon WITH "COMPANY"
      RETURN
CASE RAT("PLT ",UPPER(unitname))<>0
      REPLACE echelon WITH "COMPANY"
      RETURN
CASE RAT("PLT,",UPPER(unitname))<>0
      REPLACE echelon WITH "COMPANY"
      RETURN
CASE RAT("TEAM",UPPER(unitname))<>0
      REPLACE echelon WITH "COMPANY"
      RETURN
CASE RAT("TM ",UPPER(unitname))<>0
      REPLACE echelon WITH "COMPANY"
      RETURN
CASE RAT("TM,",UPPER(unitname))<>0
      REPLACE echelon WITH "COMPANY"
      RETURN
CASE RAT("COMMAND",UPPER(unitname))<>0
      REPLACE echelon WITH "COMPANY"
      RETURN
CASE RAT("UNIT",UPPER(unitname))<>0
      REPLACE echelon WITH "COMPANY"
      RETURN
CASE RAT("SQUAD",UPPER(unitname))<>0
      REPLACE echelon WITH "COMPANY"
      RETURN
CASE RAT("CTR",UPPER(unitname))<>0
      REPLACE echelon WITH "COMPANY"
      RETURN
CASE RAT("DETACHMENT",UPPER(unitname))<>0
      REPLACE echelon WITH "COMPANY"
      RETURN
CASE RAT("DET ",UPPER(unitname))<>0
      REPLACE echelon WITH "COMPANY"
      RETURN
CASE RAT("DET,",UPPER(unitname))<>0
      REPLACE echelon WITH "COMPANY"
      RETURN
CASE RAT("DETACH",UPPER(unitname))<>0
      REPLACE echelon WITH "COMPANY"
      RETURN
CASE RAT("ORGANIZATION",UPPER(unitname))<>0
      REPLACE echelon WITH "COMPANY"
      RETURN
CASE RAT("CSC",UPPER(unitname))<>0
      REPLACE echelon WITH "COMPANY"
      RETURN
CASE RAT("BN",UPPER(unitname))<>0
      REPLACE echelon WITH "BATTALION"
      RETURN
CASE RAT("BATTALION",UPPER(unitname))<>0
      REPLACE echelon WITH "BATTALION"
      RETURN
CASE RAT("SQDN",UPPER(unitname))<>0
```

```
      REPLACE echelon WITH "BATTALION"
      RETURN
CASE RAT("SQUADRON",UPPER(unitname))<>0
      REPLACE echelon WITH "BATTALION"
      RETURN
CASE RAT("DIVISION",UPPER(unitname))<>0
      REPLACE echelon WITH "DIVISION"
      RETURN
CASE RAT("DIV",UPPER(unitname))<>0
      REPLACE echelon WITH "DIVISION"
      RETURN
CASE RAT("ACR",UPPER(unitname))<>0
      REPLACE echelon WITH "DIVISION"
      RETURN
CASE RAT("REGIMENT",UPPER(unitname))<>0
      REPLACE echelon WITH "DIVISION"
      RETURN
CASE RAT("HOSPITAL",UPPER(unitname))<>0
      REPLACE echelon WITH "DIVISION"
      RETURN
      RETURN
ENDCASE

*: EOF: ECHELON.PRG
```

```
*:****************************************************************
*:
*:          Program:  D:\SURE\EDITHELP.PRG
*:
*:        Called by:  HELP_PRO              (procedure in SURE.PRG)
*:
*:            Calls:  SETBROWS.PRG
*:                 :  ENDBROWS.PRG
*:                 :  MEMOPAGR()            (function  in MEMOPAGR.PRG)
*:
*:        Documented 04/16/94 at 11:02                 SNAP!   version 5.00
*:****************************************************************
* helpndx.prg
SELECT 39
SET ORDER TO subject
GO TOP
ACTIVATE WINDOW browser
DO setbrows
BROWSE FIELDS help->program,help->notes,help->subject
DO endbrows
DEACTIVATE WINDOW browser
? memopagr(help->notes,1,1,22,78)
@ 22,0 SAY "   "
SET ORDER TO PROGRAM
*: EOF: EDITHELP.PRG
```

```
*:**********************************************************************
*:
*: Procedure file: D:\SURE\EDITORDE.PRG
*:
*:   Procs & Fncts: STRE_PRO
*:               : ORDT_PRO
*:               : GETORDER
*:               : EDITPLEN
*:               : 1STRALL
*:               : 1STRONE
*:               : 1STR1PRI
*:               : PHASEBYS
*:               : VIEWSTR
*:
*:         Set by: ORDE_PRO          (procedure in SURE.PRG)
*:
*:          Calls: STARTOR           (procedure in CREATEOR.PRG)
*:               : TOMP_PRO          (procedure in CREATEOR.PRG)
*:               : STRE_PRO          (procedure in EDITORDE.PRG)
*:               : ORDT_PRO          (procedure in EDITORDE.PRG)
*:
*:         Documented 04/16/94 at 11:01                SNAP!  version 5.00
*:**********************************************************************
* editorde.prg
STORE PROGRAM() TO progvar
SET PROCEDURE TO instorde
STORE 0 TO numberva
STORE 0 TO numberst
STORE SPACE(8) TO delnumbe
STORE SPACE(8) TO pnumberv
STORE SPACE(50) TO namevar
STORE 1 TO statusvar
STORE 0 TO nextpop
STORE SPACE(60) TO seekvar
STORE 0 TO diffvar
* Variable used to add to taskorg after number of Phases change +
STORE SPACE(8) TO addnumva
STORE SPACE(60) TO tnameva
STORE 0 TO phaseva
STORE SPACE(20) TO pronamva
STORE 100 TO strengtv
CLEAR
DO startor
* Popup for Mission Profile/Task Organization assignment choices
DEFINE POPUP tomp_pop FROM 1,3
DEFINE BAR 1 OF tomp_pop PROMPT ;
    "Assign one Parameter Set to all Task Organizations and Phases"
DEFINE BAR 2 OF tomp_pop PROMPT ;
    "Assign a Parameter Set to one Task Organization for all Phases"
DEFINE BAR 3 OF tomp_pop PROMPT ;
    "Assign a Parameter Set to all Task Organizations for a single Phase"
DEFINE BAR 4 OF tomp_pop PROMPT ;
    "Assign Parameter Sets to a Task Organization Phase by Phase"
DEFINE BAR 5 OF tomp_pop PROMPT ;
    "View this Orders Task Organizations and Parameter Sets by Phase"
DEFINE BAR 6 OF tomp_pop PROMPT "Return to Previous Menu"
ON SELECTION POPUP tomp_pop DO tomp_pro
ON BAR 1 OF tomp_pop DO tomp_bar1
ON BAR 2 OF tomp_pop DO tomp_bar2
ON BAR 3 OF tomp_pop DO tomp_bar3
```

```
ON BAR 4 OF tomp_pop DO tomp_bar4
ON BAR 5 OF tomp_pop DO tomp_bar5
ON BAR 6 OF tomp_pop DO tomp_bar6
* Popup for Strength Percentages
DEFINE POPUP stre_pop FROM 1,3
DEFINE BAR 1 OF stre_pop PROMPT ;
   "Assign one Strength to all Task Organizations and Phases this Order"
DEFINE BAR 2 OF stre_pop PROMPT ;
   "Assign a Strength to one Task Organization for all Phases this Order"
DEFINE BAR 3 OF stre_pop PROMPT ;
   "Assign a Strength to all Task Organizations for a single Phase this Order"
DEFINE BAR 4 OF stre_pop PROMPT ;
   "Assign Strengths to a Task Organization Phase by Phase for this Order"
DEFINE BAR 5 OF stre_pop PROMPT ;
   "View this Orders Task Organizations and Strengths by Phase"
DEFINE BAR 6 OF stre_pop PROMPT "Return to Previous Menu"
ON SELECTION POPUP stre_pop DO stre_pro
ON BAR 1 OF stre_pop DO stre_bar1
ON BAR 2 OF stre_pop DO stre_bar2
ON BAR 3 OF stre_pop DO stre_bar3
ON BAR 4 OF stre_pop DO stre_bar4
ON BAR 5 OF stre_pop DO stre_bar5
ON BAR 6 OF stre_pop DO stre_bar6
* Main Popup
DEFINE POPUP ordt_pop FROM 1,23
DEFINE BAR 1 OF ordt_pop PROMPT "Get Order"
DEFINE BAR 2 OF ordt_pop PROMPT "View/Delete Task Organizations"
DEFINE BAR 3 OF ordt_pop PROMPT "Assign Task Organizations"
DEFINE BAR 4 OF ordt_pop PROMPT "Assign Mission Parameter Sets to Task
Organization Phases"
DEFINE BAR 5 OF ordt_pop PROMPT "Assign Strength Percentages to Task
Organization Phases"
DEFINE BAR 6 OF ordt_pop PROMPT "Edit Order Description, Phases, and Phase
Lengths"
DEFINE BAR 7 OF ordt_pop PROMPT "Save and Edit Another"
DEFINE BAR 8 OF ordt_pop PROMPT "Return to Main Menu"
ON SELECTION POPUP ordt_pop DO ordt_pro
ON BAR 1 OF ordt_pop DO ordt_bar1
ON BAR 2 OF ordt_pop DO ordt_bar2
ON BAR 3 OF ordt_pop DO ordt_bar3
ON BAR 4 OF ordt_pop DO ordt_bar4
ON BAR 5 OF ordt_pop DO ordt_bar5
ON BAR 6 OF ordt_pop DO ordt_bar6
ON BAR 7 OF ordt_pop DO ordt_bar7
ON BAR 8 OF ordt_pop DO ordt_bar8
DO WHILE PROMPT()<>"Return to Main Menu"
   ACTIVATE POPUP ordt_pop
   IF nextpop=1
      ACTIVATE POPUP tomp_pop
      nextpop=0
      ACTIVATE POPUP ordt_pop
   ENDIF (nextpop=1)
   IF nextpop=2
      SELECT 18
      SET FILTER TO ordertas->number=orderlst->number
      GO TOP
      ACTIVATE POPUP stre_pop
      nextpop=0
      ACTIVATE POPUP ordt_pop
   ENDIF (nextpop=2)
```

161

```
ENDDO (PROMPT()<>"Return to Main Menu")
*!************************************************************************
*!
*!        Procedure: ORDT_PRO
*!
*!        Called by: EDITORDE.PRG
*!
*!            Calls: STARTOR              (procedure in CREATEOR.PRG)
*!                 : GETORDER             (procedure in EDITORDE.PRG)
*!                 : VIEWORDE             (procedure in CREATEOR.PRG)
*!                 : DISPORD2             (procedure in CREATEOR.PRG)
*!                 : ASSIGNTA             (procedure in CREATEOR.PRG)
*!                 : EDITPLEN             (procedure in EDITORDE.PRG)
*!                 : ENDORDR              (procedure in CREATEOR.PRG)
*!
*!************************************************************************
PROCEDURE ordt_pro
DO CASE
CASE BAR()=1
   IF statusvar<>1
      DO startor
   ENDIF (statusvar<>1)
   DO getorder
CASE BAR()=2
   DO vieworde
   IF statusvar=2
      DO dispord2
   ENDIF (statusvar=2)
CASE BAR()=3
   DO assignta
   IF statusvar=2
      DO dispord2
   ENDIF (statusvar=2)
CASE BAR()=4
   IF statusvar=1
      ACTIVATE WINDOW small
      ? CHR(7)
      @ 1,2 SAY "You must select an Order first!"
      @ 3,3 SAY "Press any key..."
      WAIT ""
      DEACTIVATE WINDOW small
      CLEAR
   ELSE
      nextpop=1
      DEACTIVATE POPUP
   ENDIF (statusvar=1)
   IF statusvar=2
      DO dispord2
   ENDIF (statusvar=2)
CASE BAR()=5
   IF statusvar=1
      ACTIVATE WINDOW small
      ? CHR(7)
      @ 1,2 SAY "You must select an Order first!"
      @ 3,3 SAY "Press any key..."
      WAIT ""
      DEACTIVATE WINDOW small
      CLEAR
   ELSE
      nextpop=2
```

```
                DEACTIVATE POPUP
        ENDIF (statusvar=1)
        IF statusvar=2
            DO dispord2
        ENDIF (statusvar=2)
    CASE BAR()=6
        DO editplen
        IF statusvar=2
            DO dispord2
        ENDIF (statusvar=2)
    CASE BAR()=7
        DO startor
        IF statusvar=2
            DO dispord2
        ENDIF (statusvar=2)
    CASE BAR()=8
        DO endordr
    ENDCASE
    * CASE BAR()=1
    *!****************************************************************
    *!
    *!        Procedure: GETORDER
    *!
    *!        Called by: ORDT_PRO              (procedure in EDITORDE.PRG)
    *!                 : DORD_PRO              (procedure in DELORDE.PRG)
    *!                 : ORDR_PRO              (procedure in ORDEREP1.PRG)
    *!
    *!            Calls: SETBROWS.PRG
    *!                 : ENDBROWS.PRG
    *!                 : DISPORD2              (procedure in CREATEOR.PRG)
    *!                 : VIEWORDE              (procedure in CREATEOR.PRG)
    *!                 : DISPORD               (procedure in CREATEOR.PRG)
    *!                 : DISPORD3              (procedure in ORDEREP1.PRG)
    *!                 : CALCPOL.PRG
    *!                 : CALCAMMO.PRG
    *!                 : CALCOTHE.PRG
    *!
    *!****************************************************************
    PROCEDURE getorder
    SELECT 3
    * Make sure there are some records to view
    IF RECCOUNT()=0
        ACTIVATE WINDOW small
        ? CHR(7)
        @ 1,2 SAY "You must have at"
        @ 2,2 SAY "least one Order!"
        @ 3,3 SAY "Press any key..."
        WAIT ""
        DEACTIVATE WINDOW small
        CLEAR
        RETURN
    ENDIF (RECCOUNT()=0)
    ACTIVATE WINDOW browser
    GO TOP
    DO setbrows
    BROWSE NOMENU NOEDIT NOAPPEND NODELETE FIELDS NUMBER,name, descriptio FREEZE
    name COMPRESS
    DO endbrows
    DEACTIVATE WINDOW browser
    SET CURSOR OFF
```

```
 CLEAR
 DO WHILE LASTKEY()<>23
    DO dispord2
    IF RECCOUNT()=1
       @ 23,30 SAY "Crtl-W when Done"
    ELSE
       @ 23,15 SAY "Page-Up or Page-Down to View, Ctrl-W when Done"
    ENDIF (RECCOUNT()=1)
    IF LASTKEY()=3
       SKIP
       IF EOF()
          GO TOP
       ENDIF (EOF())
    ENDIF (LASTKEY()=3)
    IF LASTKEY()=18
       SKIP -1
       IF BOF()
          GO BOTTOM
       ENDIF (BOF())
    ENDIF (LASTKEY()=18)
    WAIT ""
 ENDDO (LASTKEY()<>23)
 SET CURSOR ON
 SET MESSAGE TO
 DO dispord2
 statusvar=2
 * Restrict ordertas.dbf to records in this Order
 SELECT 18
 SET ORDER TO
 SET FILTER TO NUMBER=orderlst->number
 GO TOP
 * CASE BAR()=2
 *!*********************************************************************
 *!
 *!        Procedure: VIEWORDE
 *!
 *!        Called by: ORDS_PRO           (procedure in CREATEOR.PRG)
 *!                 : ORDT_PRO           (procedure in EDITORDE.PRG)
 *!                 : GETORDER           (procedure in EDITORDE.PRG)
 *!
 *!            Calls: SETBROWS.PRG
 *!                 : ENDBROWS.PRG
 *!                 : DISPORD2           (procedure in CREATEOR.PRG)
 *!                 : DISPORD3           (procedure in ORDEREP1.PRG)
 *!
 *!*********************************************************************
 PROCEDURE vieworde
 IF statusvar=1
    ACTIVATE WINDOW small
    ? CHR(7)
    @ 1,2 SAY "You must choose an Order first!"
    @ 3,3 SAY "Press any key..."
    WAIT ""
    DEACTIVATE WINDOW small
    CLEAR
    RETURN
 ENDIF (statusvar=1)
 * Restrict view to Task Orgs in this Order
 SELECT 18
 REPLACE ALL changestat WITH .n. FOR changestat
```

164

```
       SET FILTER TO NUMBER=orderlst->number
       GO TOP
       * Make sure there are some records to view
       IF EOF()
          ACTIVATE WINDOW small
          ? CHR(7)
          @ 1,2 SAY "You must have at"
          @ 2,2 SAY "least one Task Org!"
          @ 3,3 SAY "Press any key..."
          WAIT ""
          DEACTIVATE WINDOW small
          CLEAR
          RETURN
       ENDIF (EOF())
       SET ORDER TO uniqueta
       ACTIVATE WINDOW browser
       GO TOP
       SET MESSAGE TO "Enter Y to Delete Task Organization from this order, CRTL-W
       when done."
       REPLACE ALL changestat WITH .F. FOR changestat
       GO TOP
       BROWSE NOMENU NOAPPEND NODELETE FIELDS changestat,taskorgnam FREEZE changestat
       COMPRESS
       * Need to Delete from ordertas.dbf too!
       SET FILTER TO changestat
       GO TOP
       IF .NOT. EOF()
          SET FILTER TO
          SET ORDER TO
          GO TOP
          DO WHILE .T.
             GO TOP
             LOCATE FOR changestat .AND. .NOT. DELETED()
             IF .NOT. FOUND()
                EXIT
             ENDIF (.NOT. FOUND())
             namevar=taskorgnam
             delnumbe=NUMBER
             DELETE FOR NUMBER=delnumbe .AND. taskorgnam=namevar
          ENDDO (.T.)
          SET FILTER TO
          PACK
       ENDIF (.NOT. EOF())
       SET FILTER TO
       SET ORDER TO taskorgnam
       DEACTIVATE WINDOW browser
       CLEAR
       SET MESSAGE TO
       * CASE BAR()=3
*!*********************************************************************
*!
*!        Procedure: ASSIGNTA
*!
*!        Called by: ORDS_PRO            (procedure in CREATEOR.PRG)
*!                 : ORDT_PRO            (procedure in EDITORDE.PRG)
*!
*!            Uses: TTEMP.DBF
*!                : TEMP.DBF
*!
*!*********************************************************************
```

165

```
PROCEDURE assignta
IF statusvar=1
   ACTIVATE WINDOW small
   ? CHR(7)
   @ 1,2 SAY "You must choose an Order first!"
   @ 3,3 SAY "Press any key..."
   WAIT ""
   DEACTIVATE WINDOW small
   CLEAR
   RETURN
ENDIF (statusvar=1)
* Select Task Organizations for this Order
* Make sure order has 1 or more phases
SELECT 3
IF phases=0
   ACTIVATE WINDOW small
   ? CHR(7)
   @ 1,1 SAY "Order has ZERO phases,you will have to"
   @ 2,1 SAY "add some from the Edit Order Screen."
   @ 3,3 SAY "Press any key..."
   WAIT ""
   DEACTIVATE WINDOW small
   CLEAR
   RETURN
ENDIF (phases=0)
STORE phases TO phasevar
* prepare taskorg so user can only add task forces not part of the order
SELECT 1
SET ORDER TO taskorgnam
REPLACE ALL changestat WITH .n.
GO TOP
REPLACE ALL changesta2 WITH .n.
GO TOP
SELECT 18
GO TOP
DO WHILE .NOT. EOF()
   seekvar=ordertas->taskorgnam
   SELECT 1
   SEEK seekvar
   IF FOUND()
      REPLACE changestat WITH .y.
   ELSE
      GO TOP
   ENDIF (FOUND())
   SELECT 18
   SKIP
ENDDO (.NOT. EOF())
GO TOP
SELECT 1
SET FILTER TO .NOT. changestat
* User should only see each Task Org name listed once
SET ORDER TO uniqueta
SET MESSAGE TO "Enter Y for each Task Organization in this Order. Ctrl-W when
Done"
ACTIVATE WINDOW browser
GO TOP
BROWSE NOAPPEND NODELETE NOMENU FIELDS changesta2, taskorgnam FREEZE changesta2
COMPRESS
DEACTIVATE WINDOW browser
SET FILTER TO
```

```
SET ORDER TO taskorgnam
LOCATE FOR changesta2
IF FOUND()
   * Mark all changestat for selected Task Orgs
   COPY TO ttemp FOR changesta2
   SELECT 15
   USE ttemp
   DO WHILE .NOT. EOF()
      SELECT 1
      REPLACE ALL changestat WITH .T. FOR taskorg->taskorgnam=ttemp->taskorgnam
      GO TOP
      SELECT 15
      SKIP
   ENDDO (.NOT. EOF())
   SELECT 15
   USE
   SELECT 1
   COPY TO ttemp FOR changestat .AND. changesta2
   SELECT 18
   COPY STRUCTURE TO temp
   * Create one record for each Task Organization for each Phase of the Order
   SELECT 11
   USE temp
   STORE 1 TO countvar
   SET MESSAGE TO "Please wait, working..."
   DO WHILE .T.
      APPEND FROM ttemp
      REPLACE ALL phase WITH countvar FOR phase=0
      countvar=countvar+1
      IF countvar>phasevar
         EXIT
      ENDIF (countvar>phasevar)
   ENDDO (.T.)
   REPLACE ALL NUMBER WITH orderlst->number
   * Initialize Task Orgs for this order at 100 %
   REPLACE ALL strength WITH 100
   SELECT 11
   USE
   SELECT 18
   APPEND FROM temp
   SELECT 1
   REPLACE ALL changestat WITH .F. FOR changestat
   SET MESSAGE TO
ENDIF (FOUND())
KEYBOARD "{DNARROW}"
* CASE BAR()=4
*!*****************************************************************
*!
*!      Procedure: ENDORDR
*!
*!      Called by: ORDS_PRO          (procedure in CREATEOR.PRG)
*!               : ORDT_PRO          (procedure in EDITORDE.PRG)
*!
*!*****************************************************************
PROCEDURE endordr
SELECT 18
SET ORDER TO profilenam
SEEK " "
IF FOUND()
   STORE "YES" TO yesnovar
```

167

```
    ACTIVATE WINDOW small
    ? CHR(7)
    @ 1,1 SAY "You have not assigned Parameter Sets to"
    @ 2,1 SAY "all units. Cancel exit? "
    @ ROW(),COL() GET yesnovar
    READ
    IF yesnovar="YES"
        DEACTIVATE WINDOW small
        RETURN
    ELSE
        DEACTIVATE WINDOW small
    ENDIF (yesnovar="YES")
ENDIF (FOUND())
CLOSE DATA
DEACTIVATE POPUP
RETURN
*!*******************************************************************
*!
*!        Procedure: STARTOR
*!
*!        Called by: CREATEOR.PRG
*!                 : EDITORDE.PRG
*!                 : ORDS_PRO            (procedure in CREATEOR.PRG)
*!                 : ORDT_PRO            (procedure in EDITORDE.PRG)
*!
*!            Calls: STARTHLP.PRG
*!
*!             Uses: CPROFILE.DBF
*!                 : TASKORG.DBF
*!                 : ORDERTAS.DBF
*!                 : ORDERLST.DBF
*!
*!        MDX files: CPROFILE.MDX
*!                 : TASKORG.MDX
*!                 : ORDERTAS.MDX
*!                 : ORDERLST.MDX
*!
*!*******************************************************************
PROCEDURE startor
STORE 1 TO statusvar
DO starthlp
SELECT 4
USE cprofile
SELECT 1
USE taskorg ORDER uniqueta
SELECT 18
USE ordertas
SELECT 3
USE orderlst
*!*******************************************************************
*!
*!        Procedure: DISPORD
*!
*!        Called by: CREATEO              (procedure in CREATEOR.PRG)
*!                 : GETORDER             (procedure in EDITORDE.PRG)
*!
*!*******************************************************************
PROCEDURE dispord
@ 17,2 TO 17,78
@ 19,2 SAY "Order Number:   "
```

```
@ ROW(),COL() SAY pnumberv
@ 21,2 SAY "  Order Name:   "
@ ROW(),COL() SAY namevar
*!*******************************************************************
*!
*!        Procedure: DISPORD2
*!
*!        Called by: VIEWORDE           (procedure in CREATEOR.PRG)
*!                 : ORDT_PRO           (procedure in EDITORDE.PRG)
*!                 : GETORDER           (procedure in EDITORDE.PRG)
*!
*!*******************************************************************
PROCEDURE dispord2
SELECT 3
@ 20,1 SAY "Order Number:   "
@ ROW(),COL() SAY NUMBER
@ 21,2 SAY " Order Name:   "
@ ROW(),COL() SAY name
@ 22,2 SAY "Description:   "
@ ROW(), COL() SAY descriptio
@ 23,2 SAY "Number of Phases:   "
@ ROW(), COL() SAY LTRIM(STR(phases))
* CASE BAR()=6             '
*!*******************************************************************
*!
*!        Procedure: EDITPLEN
*!
*!        Called by: ORDT_PRO           (procedure in EDITORDE.PRG)
*!
*!*******************************************************************
PROCEDURE editplen
CLEAR
IF statusvar=1
   ACTIVATE WINDOW small
   ? CHR(7)
   @ 1,2 SAY "You must choose an Order first!"
   @ 3,3 SAY "Press any key..."
   WAIT ""
   DEACTIVATE WINDOW small
   CLEAR   .
   RETURN
ENDIF (statusvar=1)
SET CONFIRM ON
ACTIVATE WINDOW editor
numberva=orderlst->phases
DO WHILE LASTKEY()<>23
   @ 1,1  SAY "Order No. "
   @ 1,13 SAY NUMBER PICTURE "XXXXXXXX"
   @ 3,1  SAY "Order Name: "
   @ 3,13 GET orderlst->name PICTURE
"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
   @ 5,1  SAY "Description:"
   @ 6,6  GET orderlst->descriptio PICTURE
"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
   @ 8,1  SAY "Number of Phases: "
   @ 8,19 GET orderlst->phases PICTURE "9"
   @ 10,1 SAY "Phase Lengths (Days):"
   @ ROW(),COL()+3 SAY "P-1"
   @ ROW(),COL()+1 GET orderlst->phasellen
   @ ROW(),COL()+3 SAY "P-2"
```

169

```
@ ROW(),COL()+1 GET orderlst->phase2len
@ ROW(),COL()+3 SAY "P-3"
@ ROW(),COL()+1 GET orderlst->phase3len
@ ROW(),COL()+3 SAY "P-4"
@ ROW(),COL()+1 GET orderlst->phase4len
@ 12,1
@ ROW(),COL() SAY "P-5"
@ ROW(),COL()+1 GET orderlst->phase5len
@ ROW(),COL()+3 SAY "P-6"
@ ROW(),COL()+1 GET orderlst->phase6len
@ ROW(),COL()+3 SAY "P-7"
@ ROW(),COL()+1 GET orderlst->phase7len
@ ROW(),COL()+3 SAY "P-8"
@ ROW(),COL()+1 GET orderlst->phase8len
@ ROW(),COL()+3 SAY "P-9"
@ ROW(),COL()+1 GET orderlst->phase9len
@14,2 SAY "No quantity calcuations for Phase Lengths of Zero. Ctrl-W when
Done"
   READ
   IF orderlst->phases<1
      ACTIVATE WINDOW small
      @ 1,1 SAY "Must have 1 or more Phases!"
      @ 2,1 SAY "Press any key..."
      WAIT ""
      DEACTIVATE WINDOW small
      LOOP
   ENDIF (orderlst->phases<1)
ENDDO (LASTKEY()<>23)
SET CONFIRM OFF
DEACTIVATE WINDOW editor
* Need to adjust ordertas based on changes to number of phases, if any
IF orderlst->phases<>numberva
   ****************************
   DO CASE
   CASE orderlst->phases<numberva
      SELECT 18
      DELETE FOR NUMBER=orderlst->number .AND. phase>orderlst->phases
      PACK
   CASE orderlst->phases>numberva
      SELECT 18
      * Need to keep track of how many additional phases were added
      diffvar=orderlst->phases-numberva
      SET FILTER TO
      REPLACE ALL changestat WITH .n.
      SET FILTER TO NUMBER=orderlst->number .AND. .NOT. changestat
      GO TOP
      SET ORDER TO taskorgnam
      GO TOP
      * Need to carry data to new record
      DO WHILE .NOT. EOF()
         addnumva=ordertas->number
         tnameva=ordertas->taskorgnam
         phaseva=numberva+1
         pronamva=ordertas->profilenam
         LOCATE FOR taskorgnam=tnameva .AND. phase=numberva
         strengtv=ordertas->strength
         DO WHILE .T.
            GO TOP
            LOCATE FOR taskorgnam=tnameva .AND. phase=orderlst->phases
            IF .NOT. FOUND()
```

170

```
                    APPEND BLANK
                    REPLACE NUMBER WITH addnumva
                    REPLACE taskorgnam WITH tnameva
                    REPLACE phase WITH phaseva
                    REPLACE profilenam WITH pronamva
                    REPLACE strength WITH strengtv
                    addnumva=ordertas->number
                    tnameva=ordertas->taskorgnam
                    phaseva=ordertas->phase+1
                    pronamva=ordertas->profilenam
                    strengtv=ordertas->strength
                ELSE
                    EXIT
                ENDIF (.NOT. FOUND())
            ENDDO (.T.)
            GO TOP
            REPLACE ALL changestat WITH .y. FOR taskorgnam=tnameva
            GO TOP
        ENDDO (.NOT. EOF())
        SET FILTER TO
    ENDCASE
ENDIF (orderlst->phases<>numberva)
***************************************************
* Popup for Mission Parameter Set/Task Organization assignment choices
* BAR 1 OF Tomp_pop PROMPT "Assign one Parameter Set to all Task Organizations
and Phases"
* BAR 2 OF Tomp_pop PROMPT "Assign one Parameter Set to one Task Organization
for all Phases"
* BAR 3 OF Tomp_pop PROMPT "Assign one Parameter Set to all Task Organizations
for a single Phase"
* BAR 4 OF Tomp_pop PROMPT "Assign Parameter Sets to a Task Organization Phase
by Phase"
* BAR 5 OF Tomp_pop PROMPT "View Task Organizations and Parameter Sets by
Phase"
* BAR 6 OF Tomp_pop PROMPT Return to Previous Menu"
* Procedures for tomp_pop
*!***********************************************************************
*!
*!        Procedure: TOMP_PRO
*!
*!        Called by: CREATEOR.PRG
*!                 : EDITORDE.PRG
*!
*!           Calls: 1PROALL            (procedure in CREATEOR.PRG)
*!                 : 1PROONE           (procedure in CREATEOR.PRG)
*!                 : 1PRO1PRI          (procedure in CREATEOR.PRG)
*!                 : PHASEBY           (procedure in CREATEOR.PRG)
*!                 : VIEWTOP           (procedure in CREATEOR.PRG)
*!
*!***********************************************************************
PROCEDURE tomp_pro
DO CASE
CASE BAR()=1
    DO 1proall
CASE BAR()=2
    DO 1proone
CASE BAR()=3
    DO 1pro1pri
CASE BAR()=4
    DO phaseby
```

171

```
CASE BAR()=5
    DO viewtop
CASE BAR()=6
    DEACTIVATE POPUP
ENDCASE
*CASE BAR()=1
*!*******************************************************************
*!
*!        Procedure: 1PROALL
*!
*!        Called by: TOMP_PRO              (procedure in CREATEOR.PRG)
*!
*!            Calls: FINDPRO               (procedure in CREATEOR.PRG)
*!
*!*******************************************************************
PROCEDURE 1proall
SET MESSAGE TO "Ctrl-W to Choose Parameter Set for all Task Organizations and
Phases"
DO findpro
SELECT 18
REPLACE ALL profilenam WITH cprofile->profilenam FOR
ordertas->number=orderlst->number
CLEAR
*CASE BAR()=2
*!*******************************************************************
*!
*!        Procedure: 1PROONE
*!
*!        Called by: TOMP_PRO              (procedure in CREATEOR.PRG)
*!
*!            Calls: SETBROWS.PRG
*!                 : ENDBROWS.PRG
*!                 : FINDPRO               (procedure in CREATEOR.PRG)
*!
*!*******************************************************************
PROCEDURE 1proone
* Choose Task Organization
SELECT 18
SET ORDER TO uniquetn
SET FILTER TO
SET FILTER TO NUMBER=orderlst->number
GO TOP
ACTIVATE WINDOW browser
DO setbrows
BROWSE NOAPPEND NODELETE NOEDIT NOMENU FIELDS taskorgnam COMPRESS
DO endbrows
DEACTIVATE WINDOW browser
STORE taskorgnam TO namevar
* DO findta
* Choose Parameter Set
SET MESSAGE TO "Ctrl-W to Choose Parameter Set for this Task Organizations for
all Phases"
SET FILTER TO
SET ORDER TO taskorgnam
.DO findpro
CLEAR
SELECT 18
REPLACE ALL profilenam WITH cprofile->profilenam FOR
ordertas->number=orderlst->number;
    .AND. taskorgnam=namevar
```

172

```
SET MESSAGE TO
CLEAR
*CASE BAR()=3
*!*****************************************************************
*!
*!        Procedure: 1PRO1PRI
*!
*!        Called by: TOMP_PRO              (procedure in CREATEOR.PRG)
*!
*!            Calls: FINDPRO               (procedure in CREATEOR.PRG)
*!
*!*****************************************************************
PROCEDURE 1pro1pri
SELECT 3
STORE 0 TO numbe2var
STORE phases TO numberva
SET MESSAGE TO "Ctrl-W to Choose Parameter Set for all Task Organizations and
Phases"
DO findpro
ACTIVATE WINDOW small
@ 1,1 SAY "Enter Phase Number to use selected"
@ 2,1 SAY "Parameter Set for: "
@ ROW(),COL() GET numbe2var VALID numbe2var<=numberva .AND. numbe2var>=0;
    ERROR "Must be between 0 and "+LTRIM(STR(numberva))+". 0 cancels."
@ 3,1 SAY "There are "+LTRIM(STR(numberva))+" Phases in this Order"
READ
IF numbe2var=0
    CLEAR
    ? CHR(7)
    @ 1,2 SAY "No Phase selected, action cancelled!"
    @ 3,3 SAY "Press any key..."
    WAIT ""
    DEACTIVATE WINDOW small
    CLEAR
    RETURN
ENDIF (numbe2var=0)
DEACTIVATE WINDOW small
SELECT 18
REPLACE ALL profilenam WITH cprofile->profilenam;
    FOR ordertas->number=orderlst->number .AND. phase=numbe2var
CLEAR
*CASE BAR()=4
*!*****************************************************************
*!
*!        Procedure: PHASEBY
*!
*!        Called by: TOMP_PRO              (procedure in CREATEOR.PRG)
*!
*!            Calls: VIEWTOP               (procedure in CREATEOR.PRG)
*!                 : FINDPRO               (procedure in CREATEOR.PRG)
*!
*!*****************************************************************
PROCEDURE phaseby
DO viewtop
DO findpro
SELECT 18
REPLACE profilenam WITH cprofile->profilenam
SET FILTER TO
*CASE BAR()=5
*!*****************************************************************
```

```
*!
*!          Procedure: VIEWTOP
*!
*!          Called by: TOMP_PRO                (procedure in CREATEOR.PRG)
*!                   : PHASEBY                 (procedure in CREATEOR.PRG)
*!
*!             Calls: SETBROWS.PRG
*!                   : ENDBROWS.PRG
*!
*!*********************************************************************
PROCEDURE viewtop
SELECT 18
* Put in taskorgnam+str(phase) order
SET ORDER TO taskphas
SET FILTER TO NUMBER=orderlst->number
IF NUMBER=orderlst->number
    GO RECNO()
ELSE
    GO TOP
ENDIF (number=orderlst->number)
ACTIVATE WINDOW browser
DO setbrows
BROWSE NOEDIT NOAPPEND NODELETE NOMENU FIELDS phase, profilenam, taskorgnam
DO endbrows
SET FILTER TO
SET ORDER TO taskorgnam
DEACTIVATE WINDOW browser
*!*********************************************************************
*!
*!          Procedure: DISPPROF
*!
*!          Called by: NEWPROFI                (procedure in CREATEPR.PRG)
*!                   : VIEWPROF                 (procedure in CREATEPR.PRG)
*!                   : SELEPROF                 (procedure in EDITPROF.PRG)
*!                   : CHANGEAP                 (procedure in EDITPROF.PRG)
*!                   : CHANGEAC                 (procedure in EDITPROF.PRG)
*!                   : CHANGEI                  (procedure in EDITPROF.PRG)
*!                   : CHANGEIF                 (procedure in EDITPROF.PRG)
*!                   : CHANGECI                 (procedure in EDITPROF.PRG)
*!                   : CHANGEH2                 (procedure in EDITPROF.PRG)
*!                   : CHANGEME                 (procedure in EDITPROF.PRG)
*!                   : SELECTPRO                (procedure in DELPROF.PRG)
*!                   : FINDPRO                  (procedure in CREATEOR.PRG)
*!
*!*********************************************************************
PROCEDURE dispprof
SELECT 4
@12,1 SAY "Parameter Set Name:"
@12,17 SAY profilenam PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
@13,1 SAY "Description: "
@13,17 SAY profiledes PICTURE
"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
@14,1 SAY "Area"
@14,6 SAY "Parameter Set:"
@14,17 SAY profile PICTURE "XXXXXXXXXXXXXXX"
@15, 1 SAY "Intensity:"
@15,17 SAY INTENSITY PICTURE "XXXXXXXX"
@15,27 SAY "Factor"
@15,35 SAY factor PICTURE "999.99"
@16,1 SAY "Ammo Consumption Posture: "
```

```
@ROW(),COL() SAY posture
*!**********************************************/********************
*!
*!        Procedure: FINDPRO
*!
*!        Called by: 1PROALL           (procedure in CREATEOR.PRG)
*!                 : 1PROONE           (procedure in CREATEOR.PRG)
*!                 : 1PRO1PRI          (procedure in CREATEOR.PRG)
*!                 : PHASEBY           (procedure in CREATEOR.PRG)
*!
*!           Calls: SETBROWS.PRG
*!                 : ENDBROWS.PRG
*!                 : DISPPROF          (procedure in CREATEPR.PRG)
*!
*!*****************************************************************
PROCEDURE findpro
SELECT 4
ACTIVATE WINDOW browser
GO TOP
DO setbrows
BROWSE NOMENU NOEDIT NOAPPEND NODELETE ;
    FIELDS profilenam,profiledes FREEZE profilenam COMPRESS
DO endbrows
DEACTIVATE WINDOW browser
SET CURSOR OFF
CLEAR
DO WHILE LASTKEY()<>23
    DO dispprof
    IF RECCOUNT()=1
        @ 23,30 SAY "Crtl-W when Done"
    ELSE
        @ 23,15 SAY "Page-Up or Page-Down to View, Ctrl-W when Done"
    ENDIF (RECCOUNT()=1)
    IF LASTKEY()=3
        SKIP
        IF EOF()
            GO TOP
        ENDIF (EOF())
    ENDIF (LASTKEY()=3)
    IF LASTKEY()=18
        SKIP -1
        IF BOF()
            GO BOTTOM
        ENDIF (BOF())
    ENDIF (LASTKEY()=18)
    WAIT ""
ENDDO (LASTKEY()<>23)
SET CURSOR ON
SET MESSAGE TO
*!*****************************************************************
*!
*!        Procedure: DISPTAS
*!
*!*****************************************************************
PROCEDURE disptas
@ 17,2 TO 17,78
@ 21,2 SAY "Task Organization:   "
@ ROW(),COL() SAY namevar
*!*****************************************************************
*!
```

```
*!            Procedure: FINDTA
*!
*!            Called by: EDITTASK.PRG
*!                     : COPT_PRO            (procedure in CREATETC.PRG)
*!                     : EDIT_PRO            (procedure in EDITTASK.PRG)
*!                     : DELT_PRO            (procedure in DELTASK.PRG)
*!
*!               Calls: SETBROWS.PRG
*!                     : ENDBROWS.PRG
*!                     : DISPTASK            (procedure in CREATETA.PRG)
*!
*!*******************************************************************
PROCEDURE findta
SELECT 3
STORE NUMBER TO pnumberv
SELECT 18
*STORE RECNO() TO recnovar
SET ORDER TO uniqueta
SET FILTER TO NUMBER=orderlst->number
GO TOP
ACTIVATE WINDOW browser
DO setbrows
BROWSE FIELDS taskorgnam NOMENU NOEDIT NOAPPEND NODELETE COMPRESS
DO endbrows
DEACTIVATE WINDOW browser
STORE taskorgnam TO namevar
SET FILTER TO
SET ORDER TO NUMBER
*GO recnovar
*DO disptas
*******************************************************************
*
*!*******************************************************************
*!
*!            Procedure: STRE_PRO
*!
*!            Called by: EDITORDE.PRG
*!
*!               Calls: 1STRALL            (procedure in EDITORDE.PRG)
*!                     : 1STRONE            (procedure in EDITORDE.PRG)
*!                     : 1STR1PRI           (procedure in EDITORDE.PRG)
*!                     : PHASEBYS           (procedure in EDITORDE.PRG)
*!                     : VIEWSTR            (procedure in EDITORDE.PRG)
*!
*!*******************************************************************
PROCEDURE stre_pro
* BAR 1 "Assign one Strength to all Task Organizations and Phases this Order"
* BAR 2 "Assign a Strength to one Task Organization for all Phases this Order"
* BAR 3 "Assign a Strength to all Task Organizations for a single Phase this
Order"
* BAR 4 "Assign Strengths to a Task Organization Phase by Phase for this Order"
* BAR 5 "Return to Previous Menu"
* BAR 6 "View this Orders Task Organizations and Strengths by Phase"
SELECT 18
DO CASE
CASE BAR()=1
   DO 1strall
CASE BAR()=2
   DO 1strone
CASE BAR()=3
```

```
        DO 1str1pri
CASE BAR()=4
        DO phasebys
CASE BAR()=5
        DO viewstr
CASE BAR()=6
        DEACTIVATE POPUP
ENDCASE
*CASE BAR()=1
*!*****************************************************************
*!
*!        Procedure: 1STRALL
*!
*!        Called by: STRE_PRO            (procedure in EDITORDE.PRG)
*!
*!*****************************************************************
PROCEDURE 1strall
numberst=100
SET MESSAGE TO "Strength for all Task Organizations in all Phases"
ACTIVATE WINDOW small
@ 1,1 SAY "Enter Percent Strength: "
@ ROW(),COL() GET numberst PICTURE "999" VALID numberst>=0;
        ERROR "Must be between 0 and 999"
READ
DEACTIVATE WINDOW small
REPLACE ALL ordertas->strength WITH numberst;
        FOR ordertas->number=orderlst->number
SET MESSAGE TO
CLEAR
*CASE BAR()=2
*!*****************************************************************
*!
*!        Procedure: 1STRONE
*!
*!        Called by: STRE_PRO            (procedure in EDITORDE.PRG)
*!
*!            Calls: SETBROWS.PRG
*!                 : ENDBROWS.PRG
*!
*!*****************************************************************
PROCEDURE 1strone
* Choose Task Organization
DO setbrows
BROWSE NOAPPEND NODELETE NOEDIT NOMENU FIELDS taskorgnam COMPRESS
DO endbrows
STORE taskorgnam TO namevar
numberst=100
SET MESSAGE TO "Strength for one Task Organizations in all Phases"
ACTIVATE WINDOW small
@ 1,1 SAY "Enter Percent Strength: "
@ ROW(),COL() GET numberst PICTURE "999" VALID numberst>=0;
        ERROR "Must be between 0 and 999"
READ
DEACTIVATE WINDOW small
REPLACE ALL ordertas->strength WITH numberst;
        FOR ordertas->number=orderlst->number.AND. taskorgnam=namevar
SET MESSAGE TO
CLEAR
*CASE BAR()=3
*!*****************************************************************
```

```
*!
*!        Procedure: 1STR1PRI.
*!
*!        Called by: STRE_PRO          (procedure in EDITORDE.PRG)
*!
*!****************************************************************
PROCEDURE 1str1pri
SELECT 3
STORE 0 TO numbe2var
STORE phases TO numberva
ACTIVATE WINDOW small
@ 1,1 SAY "Enter Phase Number to use selected"
@ 2,1 SAY "Parameter Set for: "
@ ROW(),COL() GET numbe2var VALID numbe2var<=numberva .AND. numbe2var>=0;
   ERROR "Must be between 0 and "+LTRIM(STR(numberva))+". 0 cancels."
@ 3,1 SAY "There are "+LTRIM(STR(numberva))+" Phases in this Order"
READ
IF numbe2var=0
   CLEAR
   ? CHR(7)
   @ 1,2 SAY "No Phase selected, action cancelled!"
   @ 3,3 SAY "Press any key..."
   WAIT ""
   DEACTIVATE WINDOW small
   CLEAR
   RETURN
ENDIF (numbe2var=0)
DEACTIVATE WINDOW small
SELECT 18
numberst=100
SET MESSAGE TO "Strength for all Task Organizations in a single Phase"
ACTIVATE WINDOW small
@ 1,1 SAY "Enter Percent Strength: "
@ ROW(),COL() GET numberst PICTURE "999" VALID numberst>=0;
   ERROR "Must be between 0 and 999"
READ
DEACTIVATE WINDOW small
REPLACE ALL ordertas->strength WITH numberst FOR
ordertas->number=orderlst->number;
   .AND. phase=numbe2var
CLEAR
*CASE BAR()=4
*!****************************************************************
*!
*!        Procedure: PHASEBYS
*!
*!        Called by: STRE_PRO          (procedure in EDITORDE.PRG)
*!
*!            Calls: SETBROW2.PRG
*!                 : ENDBROWS.PRG
*!
*!****************************************************************
PROCEDURE phasebys
ACTIVATE WINDOW browser
DO setbrow2
BROWSE NOAPPEND NODELETE NOMENU FIELDS phase, strength, taskorgnam FREEZE
strength
DO endbrows
DEACTIVATE WINDOW browser
*CASE BAR()=5
```

178

```
*!************************************************************
*!
*!        Procedure: VIEWSTR
*!
*!        Called by: STRE_PRO           (procedure in EDITORDE.PRG)
*!
*!            Calls: SETBROWS.PRG
*!                 : ENDBROWS.PRG
*!
*!************************************************************
PROCEDURE viewstr
* Put in taskorgnam+str(phase) order
ACTIVATE WINDOW browser
DO setbrows
BROWSE NOEDIT NOAPPEND NODELETE NOMENU FIELDS phase, strength, taskorgnam
DO endbrows
DEACTIVATE WINDOW browser
* Edit Phase Lengths
RETURN
*: EOF: EDITORDE.PRG
```

```
*:*******************************************************************
*:
*: Procedure file: D:\SURE\EDITPROF.PRG
*:
*:   Procs & Fncts: STARTEPR
*:                 : EDIP_PRO
*:                 : SELEPROF
*:                 : CHANGEAP
*:                 : CHANGEAC
*:                 : CHANGEI
*:                 : CHANGEIF
*:                 : CHANGECI
*:                 : CHANGEH2
*:                 : CHANGEME
*:
*:          Set by: PROF_PRO              (procedure in SURE.PRG)
*:
*:           Calls: STARTEPR              (procedure in EDITPROF.PRG)
*:                : EDIP_PRO              (procedure in EDITPROF.PRG)
*:
*:        Documented 04/16/94 at 11:01              SNAP!  version 5.00
*:*******************************************************************
* editprof.prg
STORE PROGRAM() TO progvar
CLEAR
STORE 1 TO statusvar
SET PROCEDURE TO instprof
DO startepr
DEFINE POPUP edip_pop FROM 1,14
DEFINE BAR 1 OF edip_pop PROMPT "Select Parameter Set"
DEFINE BAR 2 OF edip_pop PROMPT "Change Area Profile"
DEFINE BAR 3 OF edip_pop PROMPT "Change Ammo Consumption Posture"
DEFINE BAR 4 OF edip_pop PROMPT "Change Intensity"
DEFINE BAR 5 OF edip_pop PROMPT "Change Intensity Factor"
DEFINE BAR 6 OF edip_pop PROMPT "Change Class I Parameters"
DEFINE BAR 7 OF edip_pop PROMPT "Change Water Consumption Parameters"
DEFINE BAR 8 OF edip_pop PROMPT "Change Class VIII Medical Parameters"
DEFINE BAR 9 OF edip_pop PROMPT "Return to Main Menu"
ON SELECTION POPUP edip_pop DO edip_pro
ON BAR 1 OF edip_pop DO edip_bar1
ON BAR 2 OF edip_pop DO edip_bar2
ON BAR 3 OF edip_pop DO edip_bar3
ON BAR 4 OF edip_pop DO edip_bar4
ON BAR 5 OF edip_pop DO edip_bar5
ON BAR 6 OF edip_pop DO edip_bar6
ON BAR 7 OF edip_pop DO edip_bar7
ON BAR 8 OF edip_pop DO edip_bar8
ON BAR 9 OF edip_pop DO edip_bar9
DO WHILE PROMPT()<>"Return to Main Menu"
   ACTIVATE POPUP edip_pop
ENDDO (PROMPT()<>"Return to Main Menu")
*!*******************************************************************
*!
*!        Procedure: EDIP_PRO
*!
*!        Called by: EDITPROF.PRG
*!
*!            Calls: SELEPROF              (procedure in EDITPROF.PRG)
*!                 : CHANGEAP              (procedure in EDITPROF.PRG)
*!                 : CHANGEAC              (procedure in EDITPROF.PRG)
```

```
*!                     : CHANGEI            (procedure in EDITPROF.PRG)
*!                     : CHANGEIF           (procedure in EDITPROF.PRG)
*!                     : CHANGECI           (procedure in EDITPROF.PRG)
*!                     : CHANGEH2           (procedure in EDITPROF.PRG)
*!                     : CHANGEME           (procedure in EDITPROF.PRG)
*!                     : ENDCREP            (procedure in CREATEPR.PRG)
*!
*!****************************************************************
PROCEDURE edip_pro
DO CASE
CASE BAR()=1
   DO seleprof
CASE BAR()=2
   DO changeap
CASE BAR()=3
   DO changeac
CASE BAR()=4
   DO changei
CASE BAR()=5
   DO changeif
CASE BAR()=6
   DO changeci
CASE BAR()=7
   DO changeh2
CASE BAR()=8
   DO changeme
CASE BAR()=9
   DO endcrep
ENDCASE
* Procedures for CreP_pop
* CASE BAR() = 1
*!****************************************************************
*!
*!        Procedure: SELEPROF
*!
*!        Called by: EDIP_PRO            (procedure in EDITPROF.PRG)
*!
*!            Calls: SETBROWS.PRG
*!                 : ENDBROWS.PRG
*!                 : DISPPROF            (procedure in CREATEPR.PRG)
*!
*!****************************************************************
PROCEDURE seleprof
SELECT 4
* Make sure there are some records to view
IF RECCOUNT()=0
   ACTIVATE WINDOW small
   ? CHR(7)
   @ 1,2 SAY "You must have at"
   @ 2,2 SAY "least one Profile!"
   @ 3,3 SAY "Press any key..."
   WAIT ""
   DEACTIVATE WINDOW small
   CLEAR
   RETURN
ENDIF (RECCOUNT()=0)
ACTIVATE WINDOW browser
GO TOP
*        SET MESSAGE TO "Choose Profile to View"
DO setbrows
```

181

```
BROWSE NOMENU NOEDIT NOAPPEND NODELETE FIELDS profilenam,profiledes FREEZE
profilenam COMPRESS
DO endbrows
DEACTIVATE WINDOW browser
statusvar=2
DO dispprof
* CASE BAR() = 2
*!*************************************************************************
*!
*!        Procedure: CHANGEAP
*!
*!        Called by: EDIP_PRO          (procedure in EDITPROF.PRG)
*!
*!            Calls: SETBROWS.PRG
*!                 : ENDBROWS.PRG
*!                 : DISPPROF          (procedure in CREATEPR.PRG)
*!
*!*************************************************************************
PROCEDURE changeap
IF statusvar=1
   ACTIVATE WINDOW small
   ? CHR(7)
   @ 1,2 SAY "Choose a Parameter Set First!"
   @ 3,3 SAY "Press any key..."
   WAIT ""
   DEACTIVATE WINDOW small
   CLEAR
   RETURN
ENDIF (statusvar=1)
SELECT 7
*        SET MESSAGE TO "Choose a CASCOM Area Profile, then Ctrl-W"
DO setbrows
BROWSE NOMENU NOAPPEND NODELETE NOEDIT FIELDS profile
DO endbrows
SELECT 4
REPLACE cprofile->profile WITH profiles->profile
DO dispprof
* CASE BAR() = 3
*!*************************************************************************
*!
*!        Procedure: CHANGEAC
*!
*!        Called by: EDIP_PRO          (procedure in EDITPROF.PRG)
*!
*!            Calls: SETBROWS.PRG
*!                 : ENDBROWS.PRG
*!                 : DISPPROF          (procedure in CREATEPR.PRG)
*!
*!*************************************************************************
PROCEDURE changeac
IF statusvar=1
   ACTIVATE WINDOW small
   ? CHR(7)
   @ 1,2 SAY "Choose a Parameter Set First!"
   @ 3,3 SAY "Press any key..."
   WAIT ""
   DEACTIVATE WINDOW small
   CLEAR
   RETURN
ENDIF (statusvar=1)
```

```
SELECT 17
GO TOP
*        SET MESSAGE TO "Select a Combat Posture for Ammunition Consumption"
DO setbrows
BROWSE NOMENU NOAPPEND NODELETE NOEDIT
DO endbrows
REPLACE cprofile->posture WITH posture->posture
SET MESSAGE TO
DO dispprof
* CASE BAR() = 4
*!***********************************************************************
*!
*!        Procedure: CHANGEI
*!
*!        Called by: EDIP_PRO            (procedure in EDITPROF.PRG)
*!
*!            Calls: SETBROWS.PRG
*!                 : ENDBROWS.PRG
*!                 : DISPPROF            (procedure in CREATEPR.PRG)
*!
*!***********************************************************************
PROCEDURE changei
IF statusvar=1
   ACTIVATE WINDOW small
   ? CHR(7)
   @ 1,2 SAY "Choose a Parameter Set First!"
   @ 3,3 SAY "Press any key..."
   WAIT ""
   DEACTIVATE WINDOW small
   CLEAR
   RETURN
ENDIF (statusvar=1)
*        SET MESSAGE TO "Choose an Intensity for this Parameter Set"
SELECT 10
DO setbrows
BROWSE NOMENU NOAPPEND NODELETE NOEDIT FIELDS INTENSITY
DO endbrows
SELECT 4
REPLACE cprofile->intensity WITH intensty->intensity
REPLACE cprofile->factor WITH intensty->factor
DO dispprof
* CASE BAR() = 5
*!***********************************************************************
*!
*!        Procedure: CHANGEIF
*!
*!        Called by: EDIP_PRO            (procedure in EDITPROF.PRG)
*!
*!            Calls: DISPPROF            (procedure in CREATEPR.PRG)
*!
*!***********************************************************************
PROCEDURE changeif
ACTIVATE WINDOW small
? CHR(7)
@ 1,2 SAY "Warning! Normally CASCOM supplied!"
@ 2,2 SAY "Change with caution: "
@ ROW(),COL() GET cprofile->factor PICTURE "999.99"
READ
DEACTIVATE WINDOW small
DO dispprof
```

183

```
*  CASE BAR()=6
*!*******************************************************************
*!
*!        Procedure: CHANGECI
*!
*!        Called by: EDIP_PRO            (procedure in EDITPROF.PRG)
*!
*!            Calls: SETBROWS.PRG
*!                 : ENDBROWS.PRG
*!                 : DISPPROF             (procedure in CREATEPR.PRG)
*!
*!*******************************************************************
PROCEDURE changeci
IF statusvar=1
   ACTIVATE WINDOW small
   ? CHR(7)
   @ 1,2 SAY "Choose a Parameter Set First!"
   @ 3,3 SAY "Press any key..."
   WAIT ""
   DEACTIVATE WINDOW small
   CLEAR
   RETURN
ENDIF (statusvar=1)
* Class I
SELECT 24
DO setbrows
SET MESSAGE TO "Choose Type of Ration for Breakfast Meal, then Enter to Select"
GO TOP
LOCATE FOR class_i->ration=cprofile->cl_i_1
BROWSE NOMENU NOAPPEND NODELETE NOEDIT FIELDS ration, unit, plan_fact
REPLACE cprofile->cl_i_1 WITH class_i->ration
SET MESSAGE TO "Choose Type of Ration for Lunch Meal, then Enter to Select"
GO TOP
LOCATE FOR class_i->ration=cprofile->cl_i_2
BROWSE NOMENU NOAPPEND NODELETE NOEDIT FIELDS ration, unit, plan_fact
REPLACE cprofile->cl_i_2 WITH class_i->ration
SET MESSAGE TO "Choose Type of Ration for Dinner Meal, then Enter to Select"
GO TOP
LOCATE FOR class_i->ration=cprofile->cl_i_3
BROWSE NOMENU NOAPPEND NODELETE NOEDIT FIELDS ration, unit, plan_fact
DO endbrows
REPLACE cprofile->cl_i_3 WITH class_i->ration
SELECT 4
DO dispprof
*  CASE BAR()=7
*!*******************************************************************
*!
*!        Procedure: CHANGEH2
*!
*!        Called by: EDIP_PRO            (procedure in EDITPROF.PRG)
*!
*!            Calls: SETBROWS.PRG
*!                 : ENDBROWS.PRG
*!                 : DISPPROF             (procedure in CREATEPR.PRG)
*!
*!*******************************************************************
PROCEDURE changeh2
IF statusvar=1
   ACTIVATE WINDOW small
   ? CHR(7)
```

```
      @ 1,2 SAY "Choose a,Parameter Set First!"
      @ 3,3 SAY "Press any key..."
      WAIT ""
      DEACTIVATE WINDOW small
      CLEAR
      RETURN
ENDIF (statusvar=1)
* Select Water Parameters
SELECT 23
*        SET MESSAGE TO "Select Water Use Parameters"
DO setbrows
BROWSE NOMENU NOAPPEND NODELETE NOEDIT FIELDS gal_man_da, echelon,climate
DO endbrows
REPLACE cprofile->water_name WITH water->water_name
SELECT 4
DO dispprof
* CASE BAR()=8
*!*********************************************************************
*!
*!       Procedure: CHANGEME
*!
*!       Called by: EDIP_PRO            (procedure in EDITPROF.PRG)
*!
*!          Calls: SETBROWS.PRG
*!               : ENDBROWS.PRG
*!               : DISPPROF             (procedure in CREATEPR.PRG)
*!
*!*********************************************************************
PROCEDURE changeme
IF statusvar=1
   ACTIVATE WINDOW small
   ? CHR(7)
   @ 1,2 SAY "Choose a Parameter Set First!"
   @ 3,3 SAY "Press any key..."
   WAIT ""
   DEACTIVATE WINDOW small
   CLEAR
   RETURN
ENDIF (statusvar=1)
* Select Medical Class VIII Parameters
SELECT 22
*        SET MESSAGE TO "Select Medical Class VIII Consumption Parameters"
DO setbrows
BROWSE NOMENU NOAPPEND NODELETE NOEDIT FIELDS plan_fact,echelon,INTENSITY
DO endbrows
REPLACE cprofile->med_name WITH medical->med_name
SELECT 4
DO dispprof
* CASE BAR() = 9
*!*********************************************************************
*!
*!       Procedure: ENDCREP
*!
*!       Called by: CREP_PRO            (procedure in CREATEPR.PRG)
*!                : EDIP_PRO            ·(procedure in EDITPROF.PRG)
*!
*!*********************************************************************
PROCEDURE endcrep
CLOSE DATA
DEACTIVATE POPUP
```

```
*  CASE BAR() = 10
statusvar=1'
CLEAR
RETURN
*!*****************************************************************
*!
*!        Procedure: STARTEPR
*!
*!        Called by: EDITPROF.PRG
*!
*!            Calls: STARTHLP.PRG
*!
*!             Uses: MEDICAL.DBF
*!                 : WATER.DBF
*!                 : CLASS_I.DBF
*!                 : INTENSTY.DBF
*!                 : PROFILES.DBF
*!                 : POSTURE.DBF
*!                 : CPROFILE.DBF
*!
*!        MDX files: MEDICAL.MDX
*!                 : WATER.MDX
*!                 : CLASS_I.MDX
*!                 : INTENSTY.MDX
*!                 : PROFILES.MDX
*!                 : CPROFILE.MDX
*!
*!*****************************************************************
PROCEDURE startepr
DO starthlp
SELECT 22
USE medical
SELECT 23
USE water
SELECT 24
USE class_i
SELECT 10
USE intensty
SELECT 7
USE profiles
SELECT 17
USE posture
SELECT 4
USE cprofile
SET ORDER TO profilenam
STORE SPACE(60) TO namevar
*!*****************************************************************
*!
*!        Procedure: DISPPROF
*!
*!        Called by: NEWPROFI          (procedure in CREATEPR.PRG)
*!                 : VIEWPROF          (procedure in CREATEPR.PRG)
*!                 : SELEPROF          (procedure in EDITPROF.PRG)
*!                 : CHANGEAP          (procedure in EDITPROF.PRG)
*!                 : CHANGEAC          (procedure in EDITPROF.PRG)
*!                 : CHANGEI           (procedure in EDITPROF.PRG)
*!                 : CHANGEIF          (procedure in EDITPROF.PRG)
*!                 : CHANGECI          (procedure in EDITPROF.PRG)
*!                 : CHANGEH2          (procedure in EDITPROF.PRG)
*!                 : CHANGEME          (procedure in EDITPROF.PRG)
```

```
*!                      : SELECTPRO         (procedure in DELPROF.PRG)
*!                      : FINDPRO           (procedure in CREATEOR.PRG)
*!
*!*******************************************************************
PROCEDURE dispprof
SELECT 4
@15,1 SAY "Profile Name:"
@15,17 SAY profilenam PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
@16,1 SAY "Description: "
@16,17 SAY profiledes PICTURE
"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
@17,1 SAY "Area"
@17,6 SAY "Profile:"
@17,17 SAY profile PICTURE "XXXXXXXXXXXXXX"
@18, 1 SAY "Intensity:"
@18,17 SAY INTENSITY PICTURE "XXXXXXX"
@18,27 SAY "Factor"
@18,35 SAY factor PICTURE "999.99"
@19,1 SAY "Ammo Posture:    "
@ROW(),COL() SAY posture
@20,1 SAY "Ration Cycle:    "
@ ROW(), COL() SAY "(1) "+cl_i_1+"  (2) "+cl_i_2+"  (3) "+cl_i_3
@21,1 SAY "Water Use:       "
@ ROW(), COL() SAY water_name
@22,1 SAY "Class VIII:      "
@ ROW(), COL() SAY med_name
RETURN
*: EOF: EDITPROF.PRG
```

```
*:********************************************************************
*:
*: Procedure file: D:\SURE\EDITTASK.PRG
*:
*:   Procs & Fncts: EDIT_PRO
*:                : VIEWTASK
*:                : VIEWTAS2
*:                : UNASSIGU
*:                : UNASSIG2
*:                : ENDEDIT
*:
*:          Set by: TASK_PRO              (procedure in SURE.PRG)
*:
*:          Calls: STARTED                (procedure in DELUNIT.PRG)
*:               : FINDTA                 (procedure in CREATETC.PRG)
*:               : EDIT_PRO               (procedure in EDITTASK.PRG)
*:
*:        Documented 04/16/94 at 11:01              SNAP!  version 5.00
*:********************************************************************
*edittask.prg
STORE PROGRAM() TO progvar
* PRIVATE Variables
CLEAR
STORE 0 TO countvar
STORE SPACE(60) TO namevar
STORE SPACE(60) TO unitvar
STORE 1 TO statusvar
STORE "   " TO yesnovar
SET PROCEDURE TO insttask
DO started
IF.BOF()
   ACTIVATE WINDOW small
   ? CHR(7)
   @ 1,2 SAY "Create a Task Organization First!"
   @ 2,2 SAY "Press any key..."
   WAIT ""
   DEACTIVATE WINDOW small
   RETURN
ENDIF (BOF())
SET MESSAGE TO "Select Task Force"
DO findta
SET MESSAGE TO
DEFINE POPUP edit_pop FROM 1,5
DEFINE BAR 1 OF edit_pop PROMPT "Select Task Organization"
DEFINE BAR 2 OF edit_pop PROMPT "Assign Units - Unit Name Order"
DEFINE BAR 3 OF edit_pop PROMPT "Assign Units - SRC Order"
DEFINE BAR 4 OF edit_pop PROMPT "View Task Organization Units/Change Unit
Quantities - Unit Name Order"
DEFINE BAR 5 OF edit_pop PROMPT "View Task Organization Units/Change Unit
Quantities - SRC Order"
DEFINE BAR 6 OF edit_pop PROMPT "Un-assign Units - Unit Name Order"
DEFINE BAR 7 OF edit_pop PROMPT "Un-assign Units - SRC Order"
DEFINE BAR 8 OF edit_pop PROMPT "Return to Main Menu"
ON SELECTION POPUP edit_pop DO edit_pro
ON BAR 1 OF edit_pop DO edit_bar1
ON BAR 2 OF edit_pop DO edit_bar2
ON BAR 3 OF edit_pop DO edit_bar3
ON BAR 4 OF edit_pop DO edit_bar4
ON BAR 5 OF edit_pop DO edit_bar5
ON BAR 6 OF edit_pop DO edit_bar6
```

```
ON BAR 7 OF edit_pop DO edit_bar7
ON BAR 8 OF edit_pop DO edit_bar7
DO WHILE PROMPT()<>"Return to Main Menu"
    ACTIVATE POPUP edit_pop
ENDDO (PROMPT()<>"Return to Main Menu")
*!******************************************************************
*!
*!        Procedure: EDIT_PRO
*!
*!        Called by: EDITTASK.PRG
*!
*!             Calls: FINDTA          (procedure in CREATETC.PRG)
*!                  : ASSIGNUN        (procedure in CREATETA.PRG)
*!                  : ASSIGNU2        (procedure in CREATETA.PRG)
*!                  : VIEWTASK        (procedure in EDITTASK.PRG)
*!                  : VIEWTAS2        (procedure in EDITTASK.PRG)
*!                  : UNASSIGU        (procedure in EDITTASK.PRG)
*!                  : UNASSIG2        (procedure in EDITTASK.PRG)
*!                  : ENDEDIT         (procedure in EDITTASK.PRG)
*!
*!******************************************************************
PROCEDURE edit_pro
DO CASE
CASE BAR()=1
    DO findta
CASE BAR()=2
    DO assignun
CASE BAR()=3
    DO assignu2
CASE BAR()=4
    DO viewtask
CASE BAR()=5
    DO viewtas2
CASE BAR()=6
    DO unassigu
CASE BAR()=7
    DO unassig2
CASE BAR()=8
    DO endedit
ENDCASE
* BAR()=1
* Select Task Organization
*!******************************************************************
*!
*!        Procedure: FINDTA
*!
*!        Called by: EDITTASK.PRG
*!                  : COPT_PRO        (procedure in CREATETC.PRG)
*!                  : EDIT_PRO        (procedure in EDITTASK.PRG)
*!                  : DELT_PRO        (procedure in DELTASK.PRG)
*!
*!             Calls: SETBROWS.PRG
*!                  : ENDBROWS.PRG
*!                  : DISPTASK        (procedure in CREATETA.PRG)
*!
*!******************************************************************
PROCEDURE findta
SELECT 1
SET FILTER TO
SET ORDER TO uniqueta              && only view each task order name once
```

189

```
GO TOP
ACTIVATE WINDOW browser
DO setbrows
BROWSE FIELDS taskorgnam NOMENU NOEDIT NOAPPEND NODELETE COMPRESS
DO endbrows
DEACTIVATE WINDOW browser
STORE taskorgnam TO namevar
DO disptask
SET ORDER TO taskorgnam
statusvar=2
*  BAR()=2
*!*****************************************************************
*!
*!        Procedure: ASSIGNUN
*!
*!        Called by: CRET_PRO          (procedure in CREATETA.PRG)
*!                 : EDIT_PRO          (procedure in EDITTASK.PRG)
*!
*!            Calls: DISPTASK          (procedure in CREATETA.PRG)
*!
*!             Uses: TEMP.DBF
*!                 : TTEMP.DBF
*!
*!*****************************************************************
PROCEDURE assignun
IF statusvar=1
    ACTIVATE WINDOW small
    ? CHR(7)
    @ 1,2 SAY "Select Task Organization First!"
    @ 2,2 SAY "Press any key..."
    WAIT ""
    DEACTIVATE WINDOW small
    RETURN
ELSE
    CLEAR
    @ 1,2 SAY "Please wait, preparing database..."
    * Prepare csrclist database so user can only add units not in Task
Organization
    SELECT 2
    SET ORDER TO unitname
    REPLACE ALL taskorg WITH .n. FOR taskorg
    REPLACE ALL taskorgc WITH .n. FOR taskorgc
    GO TOP
    SELECT 1
    SET ORDER TO unitname
    SET FILTER TO
    SET FILTER TO taskorgnam=namevar
    GO TOP
    DO WHILE .T.
       * Check off unitnames
       STORE taskorg->unitname TO unitvar
       SELECT 2
       SEEK unitvar
       IF FOUND()
           REPLACE taskorg WITH .y.
       ENDIF (FOUND())
       GO TOP
       SELECT 1
       SKIP
       IF EOF()
```

190

```
            GO TOP
            SELECT 13
            EXIT
        ENDIF (EOF())
    ENDDO (.T.)
    DEACTIVATE WINDOW small
    * Just look at units not part of this task force
    SELECT 2
    SET FILTER TO
    SET FILTER TO .NOT. taskorg
    GO TOP
    IF EOF()
        CLEAR
        ACTIVATE WINDOW small
        ? CHR(7)
        @ 1,2 SAY "No Un-Assigned Units,"
        @ 2,2 SAY "No Task Force Additions!"
        @ 3,2 SAY "Press any key..."
        WAIT ""
        CLEAR
        @ 1,1 SAY "Please wait, resetting..."
        SELECT 2
        SET FILTER TO
        REPLACE ALL taskorg WITH .n. FOR taskorg
        SELECT 1
        SET FILTER TO
        @ 3,2 SAY "Press any key..."
        WAIT ""
        DEACTIVATE WINDOW small
        RETURN
    ENDIF (EOF())
    CLEAR
    * Select those to be added
    SET MESSAGE TO "Enter Y to add units to this Task Organization, then Ctrl-W"
    GO TOP
    BROWSE FIELDS taskorgc,src,unitname NOAPPEND NODELETE NOMENU FREEZE taskorgc
COMPRESS
    SET MESSAGE TO "Please wait, updating files..."
    SELECT 1
    SET FILTER TO
    COPY STRUCTURE TO temp
    SELECT 2
    * Copy user selected records
    COPY TO ttemp FOR taskorgc
    REPLACE ALL taskorgc WITH .n. FOR taskorgc
    REPLACE ALL taskorg WITH .n. FOR taskorg
    GO TOP
    * Check to see some units had been selected
    SELECT 16
    USE ttemp
    IF RECCOUNT()=0
        SELECT 16
        USE
        ACTIVATE WINDOW small
        ? CHR(7)
        @ 1,2 SAY "No Units Assigned,"
        @ 2,2 SAY "No Task Force Additions!"
        @ 3,2 SAY "Press any key..."
        WAIT ""
        DEACTIVATE WINDOW small
```

```
   ELSE
      SELECT 16
      USE
      SELECT 11
      USE temp
      APPEND FROM ttemp
      REPLACE ALL taskorgnam WITH namevar
      REPLACE ALL qty WITH 1
      SELECT 11
      USE
      SELECT 1
      APPEND FROM temp
   ENDIF (RECCOUNT()=0)
   SELECT 16
   USE ttemp
   ZAP
   SELECT 16
   USE
   SELECT 11
   USE temp
   ZAP
   SELECT 11
   USE
   SET MESSAGE TO
ENDIF (statusvar=1)
SELECT 1
SEEK namevar
DO disptask
* BAR()=3
*!*******************************************************************
*!
*!      Procedure: ASSIGNU2
*!
*!      Called by: CRET_PRO          (procedure in CREATETA.PRG)
*!               : EDIT_PRO          (procedure in EDITTASK.PRG)
*!
*!          Calls: DISPTASK          (procedure in CREATETA.PRG)
*!
*!           Uses: TEMP.DBF
*!               : TTEMP.DBF
*!
*!*******************************************************************
PROCEDURE assignu2
IF statusvar=1
   ACTIVATE WINDOW small
   ? CHR(7)
   @ 1,2 SAY "Select Task Organization First!"
   @ 2,2 SAY "Press any key..."
   WAIT ""
   DEACTIVATE WINDOW small
   RETURN
ELSE
   CLEAR
   @ 1,2 SAY "Please wait, preparing database..."
   * Prepare csrclist database so user can only add units not in Task
Organization
   SELECT 2
   SET ORDER TO unitname
   REPLACE ALL taskorg WITH .n. FOR taskorg
   REPLACE ALL taskorgc WITH .n. FOR taskorgc
```

192

```
    GO TOP
    SELECT 1
    SET ORDER TO unitname
    SET FILTER TO
    SET FILTER TO taskorgnam=namevar
    GO TOP
    DO WHILE .T.
        * Check off unitnames
        STORE taskorg->unitname TO unitvar
        SELECT 2
        SEEK unitvar
        IF FOUND()
            REPLACE taskorg WITH .y.
        ENDIF (FOUND())
        GO TOP
        SELECT 1
        SKIP
        IF EOF()
            GO TOP
            SELECT 13
            EXIT
        ENDIF (EOF())
    ENDDO (.T.)
    DEACTIVATE WINDOW small
    * Just look at units not part of this task force
    SELECT 2
    SET FILTER TO
    SET FILTER TO .NOT. taskorg
    GO TOP
    IF EOF()
        CLEAR
        ACTIVATE WINDOW small
        ? CHR(7)
        @ 1,2 SAY "No Un-Assigned Units,"
        @ 2,2 SAY "No Task Force Additions!"
        @ 3,2 SAY "Press any key..."
        WAIT ""
        CLEAR
        @ 1,1 SAY "Please wait, resetting..."
        SELECT 2
        SET FILTER TO
        REPLACE ALL taskorg WITH .n. FOR taskorg
        SELECT 1
        SET FILTER TO
        @ 3,2 SAY "Press any key..."
        WAIT ""
        DEACTIVATE WINDOW small
        RETURN
    ENDIF (EOF())
    CLEAR
    * Select those to be added
    SET MESSAGE TO "Enter Y to add units to this Task Organization, then Ctrl-W"
    GO TOP
    SET ORDER TO src
    BROWSE FIELDS taskorgc,src,unitname NOAPPEND NODELETE NOMENU FREEZE taskorgc
COMPRESS
    SET ORDER TO unitname
    SET MESSAGE TO "Please wait, updating files..."
    SELECT 1
    SET FILTER TO
```

193

```
        COPY STRUCTURE TO temp
        SELECT 2
        * Copy user selected records
        COPY TO ttemp FOR taskorgc
        REPLACE ALL taskorgc WITH .n. FOR taskorgc
        REPLACE ALL taskorg WITH .n. FOR taskorg
        GO TOP
        * Check to see some units had been selected
        SELECT 16
        USE ttemp
        IF RECCOUNT()=0
            SELECT 16
            USE
            ACTIVATE WINDOW small
            ? CHR(7)
            @ 1,2 SAY "No Units Assigned,"
            @ 2,2 SAY "No Task Force Additions!"
            @ 3,2 SAY "Press any key..."
            WAIT ""
            DEACTIVATE WINDOW small
        ELSE
            SELECT 16
            USE
            SELECT 11
            USE temp
            APPEND FROM ttemp
            REPLACE ALL taskorgnam WITH namevar
            REPLACE ALL qty WITH 1
            SELECT 11
            USE
            SELECT 1
            APPEND FROM temp
        ENDIF (RECCOUNT()=0)
        SELECT 16
        USE ttemp
        ZAP
        SELECT 16
        USE
        SELECT 11
        USE temp
        ZAP
        SELECT 11
        USE
        SET MESSAGE TO
    ENDIF (statusvar=1)
    SELECT 1
    SEEK namevar
    DO disptask
    * BAR()=4
    *!*****************************************************************
    *!
    *!        Procedure: VIEWTASK
    *!
    *!        Called by: EDIT_PRO              (procedure in EDITTASK.PRG)
    *!
    *!            Calls: SETBROW2.PRG
    *!                 : ENDBROWS.PRG
    *!                 : DISPTASK              (procedure in CREATETA.PRG)
    *!
    *!*****************************************************************
```

194

```
PROCEDURE viewtask
SELECT 1
*SET ORDER TO unitname
SET FILTER TO taskorgnam=namevar
GO TOP
DO setbrow2
BROWSE FIELDS qty,csrclist->src,unitname NOAPPEND NOMENU NODELETE COMPRESS
FREEZE qty
DO endbrows
SET KEY TO
SET ORDER TO taskorgnam
DO disptask
*  BAR()=5
*!*****************************************************************
*!
*!       Procedure: VIEWTAS2
*!
*!       Called by: EDIT_PRO            (procedure in EDITTASK.PRG)
*!
*!           Calls: SETBROW2.PRG
*!                : ENDBROWS.PRG
*!                : DISPTASK            (procedure in CREATETA.PRG)
*!
*!*****************************************************************
PROCEDURE viewtas2
SELECT 1
SET FILTER TO taskorgnam=namevar
GO TOP
DO setbrow2
SET ORDER TO src
BROWSE FIELDS qty,csrclist->src,unitname NOAPPEND NOMENU NODELETE COMPRESS
FREEZE qty
SET ORDER TO unitname
DO endbrows
SET KEY TO
SET ORDER TO taskorgnam
DO disptask
*  BAR()=6
*!*****************************************************************
*!
*!       Procedure: UNASSIGU
*!
*!       Called by: EDIT_PRO            (procedure in EDITTASK.PRG)
*!
*!           Calls: DISPTASK            (procedure in CREATETA.PRG)
*!
*!*****************************************************************
PROCEDURE unassigu
IF statusvar=1
   ACTIVATE WINDOW small
   ? CHR(7)
   @ 1,2 SAY "Select Task Organization First!"
   @ 2,2 SAY "Press any key..."
   WAIT ""
   DEACTIVATE WINDOW small
   RETURN
ENDIF (statusvar=1)
SELECT 1
SET ORDER TO taskorgnam
SET KEY TO namevar
```

195

```
SET MESSAGE TO "Enter Y for units to DELETE from this Task Organization, then
Ctrl-W"
GO TOP
BROWSE FIELDS changestat,csrclist->src,unitname NOAPPEND NODELETE NOMENU FREEZE
changestat COMPRESS
SET MESSAGE TO "Please Confirm!"
ACTIVATE WINDOW small
? CHR(7)
@ 1,2 SAY "To delete these units from the"
@ 2,2 SAY "Task Organization, type YES "
@ ROW(),COL() GET yesnovar PICTURE "!!!"
READ
SET MESSAGE TO
IF yesnovar="YES"
   DELETE FOR changestat
   PACK
ELSE
   CLEAR
   @ 1,2 SAY "Units NOT deleted!"
   @ 2,2
   REPLACE ALL changestat WITH .n. FOR changestat
   GO TOP
   WAIT
ENDIF (yesnovar="YES")
yesnovar="    "
DEACTIVATE WINDOW small
GO TOP
SEEK namevar
DO disptask
* BAR()=7
*!********************************************************************
*!
*!       Procedure: UNASSIG2
*!
*!       Called by: EDIT_PRO          (procedure in EDITTASK.PRG)
*!
*!           Calls: DISPTASK          (procedure in CREATETA.PRG)
*!
*!********************************************************************
PROCEDURE unassig2
IF statusvar=1
   ACTIVATE WINDOW small
   ? CHR(7)
   @ 1,2 SAY "Select Task Organization First!"
   @ 2,2 SAY "Press any key..."
   WAIT ""
   DEACTIVATE WINDOW small
   RETURN
ENDIF (statusvar=1)
SELECT 1
SET ORDER TO taskorgnam
SET KEY TO namevar
SET MESSAGE TO "Enter Y for units to DELETE from this Task Organization, then
Ctrl-W"
GO TOP
SET ORDER TO src
BROWSE FIELDS changestat,csrclist->src,unitname NOAPPEND NODELETE NOMENU FREEZE
changestat COMPRESS
SET ORDER TO unitname
SET MESSAGE TO "Please Confirm!"
```

196

```
ACTIVATE WINDOW small
? CHR(7)
@ 1,2 SAY "To delete these units from the"
@ 2,2 SAY "Task Organization, type YES "
@ ROW(),COL() GET yesnovar PICTURE "!!!"
READ
SET MESSAGE TO
IF yesnovar="YES"
   DELETE FOR changestat
   PACK
ELSE
   CLEAR
   @ 1,2 SAY "Units NOT deleted!"
   @ 2,2
   REPLACE ALL changestat WITH .n. FOR changestat
   GO TOP
   WAIT
ENDIF (yesnovar="YES")
yesnovar="    "
DEACTIVATE WINDOW small
GO TOP
SEEK namevar
DO disptask
* BAR()=8
*!********************************************************************
*!
*!       Procedure: ENDEDIT
*!
*!       Called by: EDIT_PRO            (procedure in EDITTASK.PRG)
*!
*!********************************************************************
PROCEDURE endedit
CLOSE DATA
DEACTIVATE POPUP
CLEAR
RETURN
*!********************************************************************
*!
*!       Procedure: STARTED
*!
*!       Called by: DELUNIT.PRG
*!                : EDITTASK.PRG
*!                : DELTASK.PRG
*!                : DELPROF.PRG
*!
*!          Calls: STARTHLP.PRG
*!
*!           Uses: CSRCLIST.DBF
*!               : CSRCLINQ.DBF
*!               : TASKORG.DBF
*!               : ORDERTAS.DBF
*!               : CPROFILE.DBF
*!
*!      MDX files: CSRCLIST.MDX
*!               : CSRCLINQ.MDX
*!               : TASKORG.MDX
*!               : ORDERTAS.MDX
*!               : CPROFILE.MDX
*!
*!********************************************************************
```

197

```
PROCEDURE started
CLEAR
DO starthlp
SELECT 2
USE csrclist ORDER unitname
SELECT 1
USE taskorg ORDER unitname
SET RELATION TO unitname INTO csrclist
SET ORDER TO taskorgnam
*!*******************************************************************
*!
*!        Procedure: DISPTASK
*!
*!        Called by: NAMETA              (procedure in CREATETA.PRG)
*!                 : ASSIGNUN            (procedure in CREATETA.PRG)
*!                 : ASSIGNU2            (procedure in CREATETA.PRG)
*!                 : FINDTA              (procedure in CREATETC.PRG)
*!                 : ASSIGNNA            (procedure in CREATETC.PRG)
*!                 : VIEWTASK            (procedure in EDITTASK.PRG)
*!                 : VIEWTAS2            (procedure in EDITTASK.PRG)
*!                 : UNASSIGU            (procedure in EDITTASK.PRG)
*!                 : UNASSIG2            (procedure in EDITTASK.PRG)
*!
*!*******************************************************************
PROCEDURE disptask
@ 17,2 TO 17,78
@ 21,2 SAY "Task Organization:   "
@ ROW(),COL() SAY namevar
*!*******************************************************************
*!
*!        Procedure: DISPUNI
*!
*!*******************************************************************
PROCEDURE dispuni
@ 19,2 SAY "      PARENT UNIT:   "
@ ROW(),COL() SAY unitvar
RETURN

*: EOF: EDITTASK.PRG
```

```
*:******************************************************************
*:
*: Procedure file: D:\SURE\EDITUNIT.PRG
*:
*:   Procs & Fncts: STARTUNI
*:                 : EDIU_PRO
*:                 : SELECTUNI
*:                 : SELECTUN2
*:                 : CHANGELI
*:                 : ADDLINCA
*:                 : DELLIN
*:                 : CHANGEEH
*:                 : ENDUNIT
*:                 : EDITSTR
*:
*:        Set by: SRC_PRO              (procedure in SURE.PRG)
*:
*:        Calls: STARTUNI             (procedure in EDITUNIT.PRG)
*:             : EDIU_PRO             (procedure in EDITUNIT.PRG)
*:
*:      Documented 04/16/94 at 11:01              SNAP!  version 5.00
*:******************************************************************
* editunit.prg
* creates master/subordinate unit relationships.
* PRIVATE Variables
** statusvar controls user attempts to do actions out of sequence
STORE PROGRAM() TO progvar
STORE SPACE(9) TO srcvar
STORE 1 TO statusvar
SET PROCEDURE TO instunit
DO startuni
DEFINE POPUP ediu_pop FROM 1,0
DEFINE BAR 1 OF ediu_pop PROMPT "Select Unit - Browse Unit List"
DEFINE BAR 2 OF ediu_pop PROMPT "Select Unit - By SRC"
DEFINE BAR 3 OF ediu_pop PROMPT "Change LIN Quantities"
DEFINE BAR 4 OF ediu_pop PROMPT "Add Equipment LIN"
DEFINE BAR 5 OF ediu_pop PROMPT "Delete LIN"
DEFINE BAR 6 OF ediu_pop PROMPT "Change Unit Strength"
DEFINE BAR 7 OF ediu_pop PROMPT "Change Unit Echelon"
DEFINE BAR 8 OF ediu_pop PROMPT "Return to Main Menu"
ON SELECTION POPUP ediu_pop DO ediu_pro
ON BAR 1 OF ediu_pop DO ediu_bar1
ON BAR 2 OF ediu_pop DO ediu_bar2
ON BAR 3 OF ediu_pop DO ediu_bar3
ON BAR 4 OF ediu_pop DO ediu_bar4
ON BAR 5 OF ediu_pop DO ediu_bar5
ON BAR 6 OF ediu_pop DO ediu_bar6
ON BAR 7 OF ediu_pop DO ediu_bar7
ON BAR 8 OF ediu_pop DO ediu_bar8
DO WHILE PROMPT()<>"Return to Main Menu"
   ACTIVATE POPUP ediu_pop
ENDDO (PROMPT()<>"Return to Main Menu")
* ON SELECTION POPOP EdiU_pop DO EdiU_pro
*!******************************************************************
*!
*!       Procedure: EDIU_PRO
*!
*!       Called by: EDITUNIT.PRG
*!
*!          Calls: SELECTUNI           (procedure in EDITUNIT.PRG)
```

199

```
*!                    : SELECTUN2           (procedure in EDITUNIT.PRG)
*!                    : CHANGELI            (procedure in EDITUNIT.PRG)
*!                    : ADDLINCA            (procedure in EDITUNIT.PRG)
*!                    : DELLIN              (procedure in EDITUNIT.PRG)
*!                    : EDITSTR             (procedure in EDITUNIT.PRG)
*!                    : CHANGEEH            (procedure in EDITUNIT.PRG)
*!                    : ENDUNIT             (procedure in EDITUNIT.PRG)
*!
*!*******************************************************************
PROCEDURE ediu_pro
DO CASE
   * Select Unit - Unit Name Order
CASE BAR() = 1
   DO selectuni
   * Select Unit - SRC Order
CASE BAR() = 2
   DO selectun2
   * Change LIN quantities
CASE BAR() = 3
   DO changeli
   * Add LIN from list
CASE BAR() = 4
   DO addlinca
   * Delete LIN
CASE BAR() = 5
   DO dellin
   * Edit Unit Strength
CASE BAR() = 6
   DO editstr
   * Change Unit Echelon
CASE BAR() = 7
   DO changeeh
   * Done
CASE BAR() = 8
   DO endunit
ENDCASE
*!*******************************************************************
*!
*!        Procedure: STARTUNI
*!
*!        Called by: EDITUNIT.PRG
*!                 : CUSTUNIT.PRG
*!
*!           Calls: STARTHLP.PRG
*!                 : HELP.PRG
*!                 : ENDUNIT             (procedure in EDITUNIT.PRG)
*!
*!            Uses: CSRCLIST.DBF
*!                : LINLIST.DBF
*!                : AMMORATE.DBF
*!                : POLRATE.DBF
*!                : SRCLIST.DBF
*!                : SRCLINQT.DBF
*!                : CSRCLINQ.DBF
*!                : ECHELONS.DBF
*!
*!        MDX files: CSRCLIST.MDX
*!                : LINLIST.MDX
*!                : AMMORATE.MDX
*!                : POLRATE.MDX
```

```
*!                      : SRCLIST.MDX
*!                      : SRCLINQT.MDX
*!                      : CSRCLINQ.MDX
*!                      : ECHELONS.MDX
*!
*!*******************************************************************
PROCEDURE startuni
CLOSE DATA
CLEAR
DO starthlp
ON KEY LABEL f1 DO HELP
SELECT 2
USE csrclist ORDER unitname
IF RECCOUNT()=0
    ? CHR(7)
    @ 15,10 SAY "You cannot use this part of SURE until you have created some
SRCs!"
    @ 18,10
    WAIT
    DO endunit
ELSE
    SELECT 33
    USE linlist ORDER lin
    SELECT 8
    USE ammorate ORDER lin
    SELECT 9
    USE polrate ORDER lin
    SELECT 5
    USE srclist NOUPDATE
    SELECT 6
    USE srclinqt ORDER lin NOUPDATE
    SELECT 2
    USE csrclist ORDER unitname
    *SELECT 13
    *SET RELATION TO unitname INTO csrclist
    SELECT 14
    USE csrclinq ORDER unitname
    SELECT 25
    USE echelons
    SELECT 2
ENDIF (RECCOUNT()=0)
*!*******************************************************************
*!
*!        Procedure: SELECTUNI
*!
*!        Called by: EDIU_PRO          (procedure in EDITUNIT.PRG)
*!                 : DELU_PRO          (procedure in DELUNIT.PRG)
*!
*!            Calls: SETBROWS.PRG
*!                 : ENDBROWS.PRG
*!                 : DISPUNIT          (procedure in CREATESR.PRG)
*!
*!*******************************************************************
PROCEDURE selectuni
SELECT 2
ACTIVATE WINDOW browser
DO setbrows
SET ORDER TO unitname
BROWSE NOMENU NOEDIT NODELETE NOAPPEND FIELDS src,unitname
DO endbrows
```

201

```
STORE unitname TO fieldvar
DEACTIVATE WINDOW browser
SET MESSAGE TO
statusvar=2
DO dispunit
*!*****************************************************************
*!
*!        Procedure: SELECTUN2
*!
*!        Called by: EDIU_PRO              (procedure in EDITUNIT.PRG)
*!                 : DELU_PRO              (procedure in DELUNIT.PRG)
*!
*!            Calls: SETBROWS.PRG
*!                 : ENDBROWS.PRG
*!                 : DISPUNIT              (procedure in CREATESR.PRG)
*!
*!*****************************************************************
PROCEDURE selectun2
SELECT 2
SET ORDER TO src
ACTIVATE WINDOW small
@ 1,1 SAY "Enter SRC or Return to Browse"
@ 2,5 GET srcvar
READ
DEACTIVATE WINDOW small
IF srcvar<>" "
   SET NEAR ON
   SEEK srcvar
   IF EOF()
      ? CHR(7)
      GO TOP
   ENDIF (EOF())
   SET NEAR OFF
ENDIF (srcvar<>" ")
ACTIVATE WINDOW browser
DO setbrows
BROWSE NOMENU NOEDIT NODELETE NOAPPEND FIELDS src,unitname
DO endbrows
STORE unitname TO fieldvar
SET ORDER TO unitname
DEACTIVATE WINDOW browser
SET MESSAGE TO
statusvar=2
DO dispunit
* Change LIN quantities
*!*****************************************************************
*!
*!        Procedure: CHANGELI
*!
*!        Called by: EDIU_PRO              (procedure in EDITUNIT.PRG)
*!                 : CUST_PRO              (procedure in CUSTUNIT.PRG)
*!
*!            Calls: SETBROWS.PRG
*!                 : ENDBROWS.PRG
*!                 : DISPUNIT              (procedure in CREATESR.PRG)
*!
*!*****************************************************************
PROCEDURE changeli
IF statusvar=1
   ACTIVATE WINDOW small
```

```
        ? CHR(7)
        @ 1,2 SAY "You must choose a unit first!"
        @ 2,2 SAY "Press any key..."
        WAIT ""
        DEACTIVATE WINDOW small
        RETURN
    ENDIF (statusvar=1)
    SELECT 14
    SET ORDER TO lin
    SET RELATION TO lin INTO ammorate, lin INTO polrate
    * limit csrclist access to records with the same unit name
    SET ORDER TO unitname
    SET KEY TO csrclist->unitname
    @ 0,25 SAY "POL-User"
    @ 0,55 SAY "AMMO-User"
    ACTIVATE WINDOW browser
    DO setbrows
    BROWSE NOAPPEND NODELETE NOMENU FIELDS lin,quantity,polrate->lin_nomen
    /28,ammorate->lin_nomen FREEZE quantity
    DO endbrows
    DEACTIVATE WINDOW browser
    @ 0,25 SAY "                "
    @ 0,55 SAY "                "
    SET MESSAGE TO
    SET RELATION TO
    SET KEY TO
    DO dispunit
    * Add LIN from database
    *!*************************************************************
    *!
    *!        Procedure: ADDLINCA
    *!
    *!        Called by: EDIU_PRO              (procedure in EDITUNIT.PRG)
    *!                 : CUST_PRO              (procedure in CUSTUNIT.PRG)
    *!
    *!           Calls: SETBROWS.PRG
    *!                 : ENDBROWS.PRG
    *!                 : DISPUNIT              (procedure in CREATESR.PRG)
    *!
    *!*************************************************************
    PROCEDURE addlinca
    IF statusvar=1
        ACTIVATE WINDOW small
        ? CHR(7)
        @ 1,2 SAY "You must choose a unit first!"
        @ 2,2 SAY "Press any key..."
        WAIT ""
        DEACTIVATE WINDOW small
        RETURN
    ENDIF (statusvar=1)
    SELECT 33
    SET ORDER TO lin
    *    SET RELATION TO lin INTO ammorate, lin INTO polrate
    *    SET ORDER TO uniqueli
    *    @ 0,25 SAY "POL-User"
    *    @ 0,55 SAY "AMMO-User"
    ACTIVATE WINDOW browser
    DO setbrows
    *    BROWSE FIELDS lin, polrate->lin_nomen, ammorate->lin_nomen
    BROWSE NOMENU NODELETE NOAPPEND FIELDS lin, NOMEN
```

203

```
DO endbrows
DEACTIVATE WINDOW browser
@ 0,25 SAY "           "
@ 0,55 SAY "            "
STORE lin TO field1
STORE 1 TO field2
*    STORE quantity TO field2
SET MESSAGE TO
ACTIVATE WINDOW editor
@ 2,2 SAY "Change Quantity to 0 if you do not want to add this LIN"
@ 4,10 SAY "LIN: "+field1
@ 5,10 SAY "Quantity: "
@ 5,21 GET field2
READ
SET RELATION TO
IF field2>0
   SELECT 14
   STORE src TO field3
   * Check to make sure LIN is not already included!
   SEEK field1
   IF .NOT. FOUND()
      APPEND BLANK
      REPLACE src WITH field3
      REPLACE lin WITH field1
      REPLACE quantity WITH field2
      REPLACE unitname WITH csrclist->unitname
   ENDIF (.NOT. FOUND())
   DEACTIVATE WINDOW editor
   SET KEY TO csrclist->unitname
   ACTIVATE WINDOW browser
   DO setbrows
   BROWSE FIELDS lin,quantity,src,unitname FREEZE quantity NOAPPEND
   DO endbrows
   DEACTIVATE WINDOW browser
ENDIF (field2>0)
DEACTIVATE WINDOW editor
SELECT 2
DO dispunit
* Delete LIN
*!********************************************************************
*!
*!      Procedure: DELLIN
*!
*!      Called by: EDIU_PRO          (procedure in EDITUNIT.PRG)
*!               : CUST_PRO          (procedure in CUSTUNIT.PRG)
*!
*!          Calls: SETBROWS.PRG
*!               : ENDBROWS.PRG
*!               : DISPUNIT          (procedure in CREATESR.PRG)
*!
*!********************************************************************
PROCEDURE dellin
IF statusvar=1
   ACTIVATE WINDOW small
   ? CHR(7)
   @ 1,2 SAY "You must choose a unit first!"
   @ 2,2 SAY "Press any key..."
   WAIT ""
   DEACTIVATE WINDOW small
   RETURN
```

```
ENDIF (statusvar=1)
SELECT 14
SET ORDER TO lin
SET RELATION TO lin INTO ammorate, lin INTO polrate
* limit csrclist access to records with the same unit name
SET ORDER TO unitname
SET KEY TO csrclist->unitname
@ 0,25 SAY "POL-User"
@ 0,55 SAY "AMMO-User"
ACTIVATE WINDOW browser
DO setbrows
BROWSE FIELDS lin,quantity,polrate->lin_nomen /28,ammorate->lin_nomen FREEZE
quantity NOEDIT
DO endbrows
DEACTIVATE WINDOW browser
@ 0,25 SAY "              "
@ 0,55 SAY "             "
SET MESSAGE TO
* Make sure user wants to delete LIN
ACTIVATE WINDOW small
STORE "    " TO MEMVAR
@ 1,1 SAY "LIN: "+lin
@ 2,1 SAY "POL-User "+polrate->lin_nomen
@ 3,1 SAY "AMMO-User "+ammorate->lin_nomen
@ 4,1 SAY "Type YES to Delete"
@ 4,21 GET MEMVAR PICTURE "!!!"
READ
CLEAR
IF MEMVAR="YES"
    DELETE
    PACK
    WAIT "LIN DELETED... Press Any Key"
ELSE
    WAIT "LIN NOT DELETED... Press Any Key"
ENDIF (MEMVAR="YES")
DEACTIVATE WINDOW small
SET RELATION TO
SET KEY TO
DO dispunit
* Change Unit Echelon
*!*********************************************************************
*!
*!      Procedure: CHANGEEH
*!
*!      Called by: EDIU_PRO           (procedure in EDITUNIT.PRG)
*!               : CUST_PRO           (procedure in CUSTUNIT.PRG)
*!
*!          Calls: SETBROWS.PRG
*!               : ENDBROWS.PRG
*!               : DISPUNIT           (procedure in CREATESR.PRG)
*!
*!           Uses: ECHELONS.DBF
*!
*!      MDX files: ECHELONS.MDX
*!
*!*********************************************************************
PROCEDURE changeeh
SELECT 25
USE echelons
DO setbrows
```

```
SET MESSAGE TO "No echelon estimated. Please choose unit echelon, then Enter to
Select."
ACTIVATE WINDOW browser
BROWSE NOEDIT NOMENU NOAPPEND NODELETE FIELDS echelon
DO endbrows
DEACTIVATE WINDOW browser
SELECT 2
REPLACE echelon WITH echelons->echelon
DO dispunit
* Return to previous menu
*!*****************************************************************
*!
*!        Procedure: ENDUNIT
*!
*!        Called by: STARTUNI            (procedure in EDITUNIT.PRG)
*!                 : EDIU_PRO            (procedure in EDITUNIT.PRG)
*!                 : CUST_PRO            (procedure in CUSTUNIT.PRG)
*!
*!*****************************************************************
PROCEDURE endunit
CLEAR
DEACTIVATE POPUP
CLOSE DATABASES
RETURN
* Edit Unit Strength
*!*****************************************************************
*!
*!        Procedure: EDITSTR
*!
*!        Called by: EDIU_PRO            (procedure in EDITUNIT.PRG)
*!
*!            Calls: DISPUNIT            (procedure in CREATESR.PRG)
*!
*!*****************************************************************
PROCEDURE editstr
IF statusvar=1
   ACTIVATE WINDOW small
   ? CHR(7)
   @ 1,2 SAY "You must choose a unit first!"
   @ 2,2 SAY "Press any key..."
   WAIT ""
   DEACTIVATE WINDOW small
   RETURN
ENDIF (statusvar=1)
ACTIVATE WINDOW small
SELECT 2
@ 2,2 SAY "Enter Unit Strength: "
@ 2,24 GET strength PICTURE "999999"
READ
DEACTIVATE WINDOW small
DO dispunit
* Display record information
*!*****************************************************************
*!
*!        Procedure: DISPUNIT
*!
*!        Called by: COPYSRC             (procedure in CREATESR.PRG)
*!                 : COPYCSRC            (procedure in CREATESR.PRG)
*!                 : ADDDATA             (procedure in CREATESR.PRG)
*!                 : SELECTUNI           (procedure in EDITUNIT.PRG)
```

206

```
*!                    : SELECTUN2          (procedure in EDITUNIT.PRG)
*!                    : CHANGELI           (procedure in EDITUNIT.PRG)
*!                    : ADDLINCA           (procedure in EDITUNIT.PRG)
*!                    : DELLIN             (procedure in EDITUNIT.PRG)
*!                    : CHANGEEH           (procedure in EDITUNIT.PRG)
*!                    : EDITSTR            (procedure in EDITUNIT.PRG)
*!                    : UNITDATA           (procedure in CUSTUNIT.PRG)
*!
*!*****************************************************************************
PROCEDURE dispunit
SELECT 2
@ 16,1 TO 16,78
@ 18,5 SAY "SRC:       "+csrclist->src
@ 20,5 SAY "Unit:      "+unitname
@ 21,5 SAY "Echelon: "+echelon
IF echelon="COMPANY"
    @ ROW(), COL() SAY " Level or Below"
ELSE
    @ ROW(), COL() SAY "                        "
ENDIF (echelon="COMPANY")
*: EOF: EDITUNIT.PRG
```

```
*:*****************************************************************************
*:
*:          Program: D:\SURE\ENDBROWS.PRG
*:
*:        Called by: HELPNDX.PRG
*:                  : EDITHELP.PRG
*:                  : COPYSRC          (procedure in CREATESR.PRG)
*:                  : COPYCSRC         (procedure in CREATESR.PRG)
*:                  : VIEWUNIT         (procedure in CREATESR.PRG)
*:                  : ADDDATA          (procedure in CREATESR.PRG)
*:                  : SELECTUNI        (procedure in EDITUNIT.PRG)
*:                  : SELECTUN2        (procedure in EDITUNIT.PRG)
*:                  : CHANGELI         (procedure in EDITUNIT.PRG)
*:                  : ADDLINCA         (procedure in EDITUNIT.PRG)
*:                  : DELLIN           (procedure in EDITUNIT.PRG)
*:                  : CHANGEEH         (procedure in EDITUNIT.PRG)
*:                  : FINDTA           (procedure in CREATETC.PRG)
*:                  : VIEWTASK         (procedure in EDITTASK.PRG)
*:                  : VIEWTAS2         (procedure in EDITTASK.PRG)
*:                  : NEWPROFI         (procedure in CREATEPR.PRG)
*:                  : VIEWPROF         (procedure in CREATEPR.PRG)
*:                  : SELEPROF         (procedure in EDITPROF.PRG)
*:                  : CHANGEAP         (procedure in EDITPROF.PRG)
*:                  : CHANGEAC         (procedure in EDITPROF.PRG)
*:                  : CHANGEI          (procedure in EDITPROF.PRG)
*:                  : CHANGECI         (procedure in EDITPROF.PRG)
*:                  : CHANGEH2         (procedure in EDITPROF.PRG)
*:                  : CHANGEME         (procedure in EDITPROF.PRG)
*:                  : SELECTPRO        (procedure in DELPROF.PRG)
*:                  : NEWCUSPR         (procedure in CRECUSPR.PRG)
*:                  : VIEWCUSP         (procedure in CRECUSPR.PRG)
*:                  : EDITCUSP         (procedure in CRECUSPR.PRG)
*:                  : DELCUSPR         (procedure in CRECUSPR.PRG)
*:                  : VIEWORDE         (procedure in CREATEOR.PRG)
*:                  : 1PROONE          (procedure in CREATEOR.PRG)
*:                  : VIEWTOP          (procedure in CREATEOR.PRG)
*:                  : FINDPRO          (procedure in CREATEOR.PRG)
*:                  : GETORDER         (procedure in EDITORDE.PRG)
*:                  : 1STRONE          (procedure in EDITORDE.PRG)
*:                  : PHASEBYS         (procedure in EDITORDE.PRG)
*:                  : VIEWSTR          (procedure in EDITORDE.PRG)
*:                  : GETSRC           (procedure in SYSREP.PRG)
*:                  : BROWHELP         (procedure in PRNTHELP.PRG)
*:
*:        Documented 04/16/94 at 11:02              SNAP!  version 5.00
*:*****************************************************************************
ON KEY LABEL ctrl-m
ON KEY LABEL HOME
ON KEY LABEL end
SET MESSAGE TO

*: EOF: ENDBROWS.PRG
```

```
*:*********************************************************************
*:'
*:        Program: D:\SURE\HELP.PRG
*:
*:      Called by: SURE.PRG
*:             : STARTHLP.PRG
*:             : STARTUNI              (procedure in EDITUNIT.PRG)
*:
*:          Calls: MEMOPAGR()          (function  in MEMOPAGR.PRG)
*:
*:      Documented 04/16/94 at 11:01                SNAP!  version 5.00
*:*********************************************************************
* help.prg
SET EXACT ON
IF SEEK(progvar,"HELP")
   IF WINDOW()<>" "
      ACTIVATE SCREEN
   ENDIF (WINDOW()<>" ")
   ? memopagr(help->notes,10,1,22,78)
   @ 22,0 SAY "  "
ENDIF (SEEK(progvar,"HELP"))
SET EXACT OFF
RETURN
*: EOF: HELP.PRG
```

209

```
*:**********************************************************************
*:
*:          Program: D:\SURE\HELPNDX.PRG
*:
*:          Called by: HELP_PRO                (procedure in SURE.PRG)
*:
*:            Calls: SETBROWS.PRG
*:                 : ENDBROWS.PRG
*:                 : MEMOPAGR()                (function  in MEMOPAGR.PRG)
*:
*:          Documented 04/16/94 at 11:02                 SNAP!  version 5.00
*:**********************************************************************
* helpndx.prg
SELECT 39
SET ORDER TO subject
GO TOP
ACTIVATE WINDOW browser
DO setbrows
BROWSE NOMENU NOAPPEND NODELETE NOEDIT FIELDS help->subject
DO endbrows
DEACTIVATE WINDOW browser
? memopagr(help->notes,1,1,22,78)
@ 22,0 SAY "   "
SET ORDER TO PROGRAM
*: EOF: HELPNDX.PRG
```

```
*:*****************************************************************
*:
*:  Procedure file: D:\SURE\INSTORDE.PRG
*:
*:   Procs & Fncts: ORDS_BAR1
*:                : ORDS_BAR2
*:                : ORDS_BAR3
*:                : ORDS_BAR4
*:                : ORDS_BAR5
*:                : TOMP_BAR1
*:                : TOMP_BAR2
*:                : TOMP_BAR3
*:                : TOMP_BAR4
*:                : TOMP_BAR5
*:                : TOMP_BAR6
*:                : ORDT_BAR1
*:                : ORDT_BAR2
*:                : ORDT_BAR3
*:                : ORDT_BAR4
*:                : ORDT_BAR5
*:                : ORDT_BAR6
*:                : ORDT_BAR7
*:                : ORDT_BAR8
*:                : DORD_BAR1
*:                : DORD_BAR2
*:                : DORD_BAR3
*:                : STRE_BAR1
*:                : STRE_BAR2
*:                : STRE_BAR3
*:                : STRE_BAR4
*:                : STRE_BAR5
*:                : STRE_BAR6
*:
*:        Documented 04/16/94 at 11:02            SNAP!  version 5.00
*:*****************************************************************
** instorde.prg
* "Create Order"
*!*****************************************************************
*!
*!        Procedure: ORDS_BAR1
*!
*!*****************************************************************
PROCEDURE ords_bar1
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
    You create orders with up to nine phases, then assign Task Organizations
    and Parameter Sets.  Order numbers may be a combination of numbers and
    letters, but must be unique, and cannot be changed later. After creating
    the order and assigning Task Organizations, you will assign Parameter Sets
    to each Task Organization for each Phase of the Order.
ENDTEXT
RETURN
* "View Orders"
*!*****************************************************************
*!
*!        Procedure: ORDS_BAR2
*!
*!*****************************************************************
PROCEDURE ords_bar2
```

```
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
    View previously created orders.
ENDTEXT


* "Assign Task Organizations"
*!*********************************************************************
*!
*!         Procedure: ORDS_BAR3
*!
*!*********************************************************************
PROCEDURE ords_bar3
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
    Each Order must have one or more Task Organizations assigned.
ENDTEXT


* "Assign Mission Profiles to Task Organization Phases"
*!*********************************************************************
*!
*!         Procedure: ORDS_BAR4
*!
*!*********************************************************************
PROCEDURE ords_bar4
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
    Each Task Organization must have an Parameter Set assigned for
    each Phase of the Order.
ENDTEXT


* "DONE!"
*!*********************************************************************
*!
*!         Procedure: ORDS_BAR5
*!
*!*********************************************************************
PROCEDURE ords_bar5
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
    Return to Main Menu.
ENDTEXT


* Assign Profiles
* "Assign one Profile to all Task Organizations and Phases this Order"
*!*********************************************************************
*!
*!         Procedure: TOMP_BAR1
*!
*!*********************************************************************
PROCEDURE tomp_bar1
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
    You can automatically assign a single Profile to all Task Organizations
    for all Phases of this Order.  This may be a useful way to start even
    if only a majority of Task Organizations use a common Operational Profile
```

```
        for a significant number of Phases.
ENDTEXT


* "Assign a Profile to one Task Organization for all Phases this Order"
*!*********************************************************************
*!
*!       Procedure: TOMP_BAR2
*!
*!*********************************************************************
PROCEDURE tomp_bar2
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT

ENDTEXT


* "Assign a Profile to all Task Organizations for a single Phase this Order"
*!*********************************************************************
*!
*!       Procedure: TOMP_BAR3
*!
*!*********************************************************************
PROCEDURE tomp_bar3
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
    Assign a single Profile to all Task Organizations.
ENDTEXT


*  "Assign Profiles to a Task Organization Phase by Phase for this Order"
*!*********************************************************************
*!
*!       Procedure: TOMP_BAR4
*!
*!*********************************************************************
PROCEDURE tomp_bar4
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
    Assign specific profiles to a single Task Organization Phase by Phase.
ENDTEXT


*  "View this Orders Task Organizations and Profiles by Phase"
*!*********************************************************************
*!
*!       Procedure: TOMP_BAR5
*!
*!*********************************************************************
PROCEDURE tomp_bar5
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
    View the Parameter Sets assigned to Task Organizations.
ENDTEXT


*  "Return to Previous Menu"
*!*********************************************************************
*!
*!       Procedure: TOMP_BAR6
*!
```

213

```
*!*******************************************************************
PROCEDURE tomp_bar6
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
    Return to the previous menu.
ENDTEXT


* Edit Order
* "Get Order"
*!*******************************************************************
*!
*!        Procedure: ORDT_BAR1
*!
*!*******************************************************************
PROCEDURE ordt_bar1
@ 11,1 CLEAR TO 17,79
@ 11,1
TEXT                .
    You must select an Order before making any changes.
ENDTEXT


* "View/Delete Task Organizations"
*!*******************************************************************
*!
*!        Procedure: ORDT_BAR2
*!
*!*******************************************************************
PROCEDURE ordt_bar2
@ 11,1 CLEAR TO 17,79
@ 11,1
TEXT
    View the Task Organizations assigned to an Order and, optionally, delele
    them.
ENDTEXT


* "Assign Task Organizations"
*!*******************************************************************
*!
*!        Procedure: ORDT_BAR3
*!
*!*******************************************************************
PROCEDURE ordt_bar3
@ 11,1 CLEAR TO 17,79
@ 11,1
TEXT
    Assign additional Task Organizations to an Order.  Only Task Organizations
    not currently assigned will be shown.
ENDTEXT


* "Assign Mission Parameter Sets to Task Organization Phases"
*!*******************************************************************
*!
*!        Procedure: ORDT_BAR4
*!
*!*******************************************************************
PROCEDURE ordt_bar4
@ 11,1 CLEAR TO 17,79
@ 11,1
TEXT
```

214

```
         Each Task Organization must have a Parameter Set assigned for each Phase
         of the order.  Use this option to assign or change Parameter Sets.
      ENDTEXT


      * "Assign Strength Percentages to Task Organization Phases"
      *!*********************************************************************
      *!
      *!       Procedure: ORDT_BAR5
      *!
      *!*********************************************************************
      PROCEDURE ordt_bar5
      @ 11,1 CLEAR TO 17,79
      @ 11,1
      TEXT
         The equipment and personnel of each Unit in each Task Organization are
         used by SURE as 100%.  You may change this setting for each Task
         Organization for each Phase of the Order.  Any changes you make will
         increase or decrease all logistics estimates proportionally.  For example,
         decrementing strength from 100% to 90% for a Phase will reduce all estimates
         by 10%.
      ENDTEXT


      * "Edit Order Description, Phases, and Phase Lengths"
      *!*********************************************************************
      *!
      *!       Procedure: ORDT_BAR6
      *!
      *!*********************************************************************
      PROCEDURE ordt_bar6
      @ 11,1 CLEAR TO 17,79
      @ 11,1
      TEXT
         While you may change the Order Description, the number of Phases, and the
         length of each Phase, you must also go back and assign Parameter Sets for
         each Task Organization in the Order if you increase the number of Phases.
      ENDTEXT


      * "Save and Edit Another"
      *!*********************************************************************
      *!
      *!       Procedure: ORDT_BAR7
      *!
      *!*********************************************************************
      PROCEDURE ordt_bar7
      @ 11,1 CLEAR TO 17,79
      @ 11,1
      TEXT
         Finalize order and prepare to create another one.
      ENDTEXT


      * "Return to Main Menu"
      *!*********************************************************************
      *!
      *!       Procedure: ORDT_BAR8
      *!
      *!*********************************************************************
      PROCEDURE ordt_bar8
      @ 11,1 CLEAR TO 17,79
      @ 11,1
      TEXT
```

215

```
       Return to the Main Menu.
ENDTEXT


* Delete Order
* Get Order
*!********************************************************************
*!
*!       Procedure: DORD_BAR1
*!
*!********************************************************************
PROCEDURE dord_bar1
@ 11,1 CLEAR TO 17,79
@ 11,1
TEXT
    Select an Order to Delete.
ENDTEXT


*!********************************************************************
*!
*!       Procedure: DORD_BAR2
*!
*!********************************************************************
PROCEDURE dord_bar2
* Delete Order
@ 11,1 CLEAR TO 17,79
@ 11,1
TEXT
    Delete selected Order.  If you delete an Order, you will not be
    able to undelete it and get it back!
ENDTEXT


*!********************************************************************
*!
*!       Procedure: DORD_BAR3
*!
*!********************************************************************
PROCEDURE dord_bar3
*Return to Main Menu
@ 11,1 CLEAR TO 17,79
@ 11,1
TEXT
    Return to the Main Menu.
ENDTEXT


* Strength Percentages
* BAR 1 OF stre_pop PROMPT ;
    *    "Assign one Strength to all Task Organizations and Phases this Order"
*!********************************************************************
*!
*!       Procedure: STRE_BAR1
*!
*!********************************************************************
PROCEDURE stre_bar1
@ 11,1 CLEAR TO 17,79
@ 11,1
TEXT
    If all or most Task Organizations will be at the same percent strength
    for all or most of the Phases in this Order, use this option to set a
    single strength throughout.  You may adjust individual cases afterwards
    with other options on this menu.
```

216

```
ENDTEXT                      ,
RETURN
* BAR 2 OF stre_pop PROMPT ;
    *    "Assign a Strength to one Task Organization for all Phases this Order"
*!*******************************************************************
*!
*!       Procedure: STRE_BAR2
*!
*!*******************************************************************
PROCEDURE stre_bar2
@ 11,1 CLEAR TO 17,79
@ 11,1
TEXT
    If one Task Organization will be at a particular strength for all or most
    of the Phases in this Order, use this option to set a single strength
    throughout the order for that Task Organization.
ENDTEXT
RETURN
* BAR 3 OF stre_pop PROMPT ;
    *    "Assign a Strength to all Task Organizations for a single Phase this
Order"
*!*******************************************************************
*!
*!       Procedure: STRE_BAR3
*!
*!*******************************************************************
PROCEDURE stre_bar3
@ 11,1 CLEAR TO 17,79
@ 11,1
TEXT
    If all or most Task Organizations will be at a particular strength during
    a particular Phase of this Order, use this option to set that strength for
    all Task Organizations for that Phase only.
ENDTEXT
RETURN
* BAR 4 OF stre_pop PROMPT ;
    *    "Assign Strengths to a Task Organization Phase by Phase for this Order"
*!*******************************************************************
*!
*!       Procedure: STRE_BAR4
*!
*!*******************************************************************
PROCEDURE stre_bar4
@ 11,1 CLEAR TO 17,79
@ 11,1
TEXT
    Use this option to pick one Task Organization during one Phase of this
    Order and set its strength for that Phase only.
ENDTEXT
RETURN
* BAR 5 OF stre_pop PROMPT ;
    *    "View this Orders Task Organizations and Strengths by Phase"
*!*******************************************************************
*!
*!       Procedure: STRE_BAR5
*!
*!*******************************************************************
PROCEDURE stre_bar5
@ 11,1 CLEAR TO 17,79
@ 11,1
```

```
TEXT
    View all Task Organizations and their strengths in this Order.
ENDTEXT
RETURN
* BAR 6 OF stre_pop PROMPT "Return to Previous Menu"
*!***********************************************************************
*!
*!       Procedure: STRE_BAR6
*!
*!***********************************************************************
PROCEDURE stre_bar6
@ 11,1 CLEAR TO 17,79
@ 11,1
TEXT
    Return to the previous menu.
ENDTEXT
RETURN

*: EOF: INSTORDE.PRG
```

```
*:*****************************************************************
*:
*:  Procedure file: D:\SURE\INSTPHLP.PRG
*:
*:   Procs & Fncts: PHEL_BAR1
*:               : PHEL_BAR2
*:               : PHEL_BAR3
*:
*:        Documented 04/16/94 at 11:02              SNAP!  version 5.00
*:*****************************************************************
*!*****************************************************************
*!
*!        Procedure: PHEL_BAR1
*!
*!*****************************************************************
PROCEDURE phel_bar1
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
   You can print any one of the context-sensitive Help screens.
ENDTEXT


RETURN
*!*****************************************************************
*!
*!        Procedure: PHEL_BAR2
*!
*!*****************************************************************
PROCEDURE phel_bar2
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
   You can print all of the Help screens in order.
ENDTEXT

*!*****************************************************************
*!
*!        Procedure: PHEL_BAR3
*!
*!*****************************************************************
PROCEDURE phel_bar3
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT

ENDTEXT


*: EOF: INSTPHLP.PRG
```

```
*:*****************************************************************
*:
*: Procedure file: D:\SURE\INSTPROF.PRG
*:
*:   Procs & Fncts: CREP_BAR1
*:                : CREP_BAR2
*:                : CREP_BAR3
*:                : EDIP_BAR1
*:                : EDIP_BAR2
*:                : EDIP_BAR3
*:                : EDIP_BAR4
*:                : EDIP_BAR5
*:                : EDIP_BAR6
*:                : EDIP_BAR7
*:                : EDIP_BAR8
*:                : EDIP_BAR9
*:                : DELP_BAR1
*:                : DELP_BAR2
*:                : DELP_BAR3
*:                : CRCP_BAR1
*:                : CRCP_BAR2
*:                : CRCP_BAR3
*:                : CRCP_BAR4
*:                : CRCP_BAR6
*:
*:        Documented 04/16/94 at 11:02              SNAP!  version 5.00
*:*****************************************************************
* instprof.prg Instructions for Profile submenu
* Create
*!*****************************************************************
*!
*!        Procedure: CREP_BAR1
*!
*!*****************************************************************
PROCEDURE crep_bar1
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
   You will assign a name, description, Area Profile, ammo comsumption
   posture combat intensity and ration cycle to a single Parameter Set.
   This Parameter Set can be assigned by Phase within an Order to one
   or more Task Organizations.


ENDTEXT
RETURN
* View
*!*****************************************************************
*!
*!        Procedure: CREP_BAR2
*!
*!*****************************************************************
PROCEDURE crep_bar2
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
   View previously created Parameter Sets.

ENDTEXT
RETURN
```

```
* Done
*!*************************************************************
*!
*!        Procedure: CREP_BAR3
*!
*!*************************************************************
PROCEDURE crep_bar3
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
    Return to Main Menu.

ENDTEXT
RETURN
* Edit Profile
* "Select Parameter Set"
*!*************************************************************
*!
*!        Procedure: EDIP_BAR1
*!
*!*************************************************************
PROCEDURE edip_bar1
@ 13,1 CLEAR TO 17,79
@ 13,1
TEXT
    Select a Parameter Set.

ENDTEXT
RETURN
* "Change Area Profile"
*!*************************************************************
*!
*!        Procedure: EDIP_BAR2
*!
*!*************************************************************
PROCEDURE edip_bar2
@ 13,1 CLEAR TO 17,79
@ 13,1
TEXT
    Assign a different Area Profile to the selected Parameter Set.


ENDTEXT
RETURN
* "Change Ammo Consumption Posture"
*!*************************************************************
*!
*!        Procedure: EDIP_BAR3
*!
*!*************************************************************
PROCEDURE edip_bar3
@ 13,1 CLEAR TO 17,79
@ 13,1
TEXT
    Assign a different Ammunition Consumption Posture to the selected
    Parameter Set.


ENDTEXT
RETURN
```

```
* "Change Intensity"
*!*****************************************************************
*!
*!       Procedure: EDIP_BAR4
*!
*!*****************************************************************
PROCEDURE edip_bar4
@ 13,1 CLEAR TO 17,79
@ 13,1
TEXT
   Assign a different combat Intensity to the selected Parameter Set.

ENDTEXT
RETURN
* "Change Intensity Factor"
*!*****************************************************************
*!
*!       Procedure: EDIP_BAR5
*!
*!*****************************************************************
PROCEDURE edip_bar5
@ 13,1 CLEAR TO 17,79
@ 13,1
TEXT
   Change the standard Intensity Factor.  The Intensity factor is
   normally a standard percentage based on the combat Intensity that
   is selected.  Adjusting the Intensity factor allows you to 'fine-tune'
   you usage projections, but makes them non-doctrinal.  Use cautiously!
ENDTEXT
RETURN
* "Change Class I Paramenters"
*!*****************************************************************
*!
*!       Procedure: EDIP_BAR6
*!
*!*****************************************************************
PROCEDURE edip_bar6
@ 13,1 CLEAR TO 17,79
@ 13,1
TEXT
   Three meal profiles are assigned for each 1 day length of the order.
   You can select a different ration type for each meal.

ENDTEXT
RETURN
* "Change Water Consumption Parameters"
*!*****************************************************************
*!
*!       Procedure: EDIP_BAR7
*!
*!*****************************************************************
PROCEDURE edip_bar7
@ 13,1 CLEAR TO 17,79
@ 13,1
TEXT
   Assign a different water consumption parameter to the selected
   Parameter Set.

ENDTEXT
RETURN
```

```
* "Change Class VIII Medical,Parameters"
*!*****************************************************************************
*!
*!        Procedure: EDIP_BAR8
*!
*!*****************************************************************************
PROCEDURE edip_bar8
@ 13,1 CLEAR TO 17,79
@ 13,1
TEXT
    Assign a different parametere for Class VIII consumption.

ENDTEXT
RETURN
* "DONE!"
*!*****************************************************************************
*!
*!        Procedure: EDIP_BAR9
*!
*!*****************************************************************************
PROCEDURE edip_bar9
@ 13,1 CLEAR TO 17,79
@ 13,1
TEXT
    Save changes and return to Main Menu.

ENDTEXT
RETURN
* Delete Profile
*!*****************************************************************************
*!
*!        Procedure: DELP_BAR1
*!
*!*****************************************************************************
PROCEDURE delp_bar1
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
    Select Parameter Set for deletion.  You will not be able to delete any
    Profiles that are currently used in an Order.
ENDTEXT
RETURN
* Delete Unit
*!*****************************************************************************
*!
*!        Procedure: DELP_BAR2
*!
*!*****************************************************************************
PROCEDURE delp_bar2
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
    Delete Parameter Set that you have selected.  You will not be able to get
    deleted Parameter Sets back!
ENDTEXT
RETURN
* Delete Unit
*!*****************************************************************************
*!
*!        Procedure: DELP_BAR3
```

```
*!                          ,                                        ,
*!***************************************************************************
PROCEDURE delp_bar3
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
    Return to Main Menu.
ENDTEXT
RETURN
* CUSTOM Profiles
* "Create Custom Area Profile"
*!***************************************************************************
*!
*!      Procedure: CRCP_BAR1
*!
*!***************************************************************************
PROCEDURE crcp_bar1
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
    Create a Custom Area Profile based on a previously existing one.  You
    must use a unique name for your Profile.  Use View Area Profiles to see a
    list of existing names.
ENDTEXT
RETURN
* "View Area Profiles"
*!***************************************************************************
*!
*!      Procedure: CRCP_BAR2
*!
*!***************************************************************************
PROCEDURE crcp_bar2
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
    Review existing Area Profile names and configurations.
ENDTEXT

RETURN
* "Edit Custom Area Profile"
*!***************************************************************************
*!
*!      Procedure: CRCP_BAR3
*!
*!***************************************************************************
PROCEDURE crcp_bar3
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
    You may edit Custom Profile configurations, assigning time and distance
    usage factors to various types of equipment.
ENDTEXT

RETURN
* "Return to Main Menu"
*!***************************************************************************
*!
*!      Procedure: CRCP_BAR4
*!
*!***************************************************************************
```

```
PROCEDURE crcp_bar4
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
    Return to the Main Menu.
ENDTEXT

RETURN
* "Delete Custom Area Profile"
*!*******************************************************************
*!
*!        Procedure: CRCP_BAR6
*!
*!*******************************************************************
PROCEDURE crcp_bar6
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
    You may delete Custom Profiles that are not currently used by any
    Parameter Set.  Standard Area Profiles will not be shown and cannot be
    deleted.
ENDTEXT

RETURN

*: EOF: INSTPROF.PRG
```

```
*:******************************************************************
*:
*: Procedure file: D:\SURE\INSTREPT.PRG
*:
*:  Procs & Fncts: ORDR_BAR1
*:               : ORDR_BAR2
*:               : ORDR_BAR3
*:               : ORDR_BAR4
*:               : ORDR_BAR5
*:
*:       Documented 04/16/94 at 11:02                SNAP!  version 5.00
*:******************************************************************
* Order-based Reports
* "Select Order/Calcuate Consumption"
*!******************************************************************
*!
*!       Procedure: ORDR_BAR1
*!
*!******************************************************************
PROCEDURE ordr_bar1
@ 11,1 CLEAR TO 17,79
@ 11,1
TEXT
   Select an Order to perform logistics estimates on.
ENDTEXT

*   "POL Estimate"
*!******************************************************************
*!
*!       Procedure: ORDR_BAR2
*!
*!******************************************************************
PROCEDURE ordr_bar2
@ 11,1 CLEAR TO 17,79
@ 11,1
TEXT
   Print bulk POL estimate to screen, file, or printer. Use Condensed Print
   if printing to Printer.
ENDTEXT

* "Ammo Estimate"
*!******************************************************************
*!
*!       Procedure: ORDR_BAR3
*!
*!******************************************************************
PROCEDURE ordr_bar3
@ 11,1 CLEAR TO 17,79
@ 11,1
TEXT
   Print ammunition estimate to screen, file, or printer. Use Condensed Print
   if printing to Printer.
ENDTEXT

* "Other Supply Estimates"
*!******************************************************************
*!
*!       Procedure: ORDR_BAR4
*!
*!******************************************************************
```

```
PROCEDURE ordr_bar4
@ 11,1 CLEAR TO 17,79
@ 11,1
TEXT
    Print ration, water, Class IV and Class VIII estimates to screen, file, or
    printer.  Use Condensed Print if printing to Printer.

ENDTEXT

* "Return to Main Menu"
*!**********************************************************************
*!
*!       Procedure: ORDR_BAR5
*!
*!**********************************************************************
PROCEDURE ordr_bar5
@ 11,1 CLEAR TO 17,79
@ 11,1
TEXT
    Return to Main Menu.
ENDTEXT


*: EOF: INSTREPT.PRG
```

```
*:******************************************************************
*:
*: Procedure file: D:\SURE\INSTRUCT.PRG
*:
*:  Procs & Fncts: SURECSR1
*:                : SURECSR2
*:                : SURECSR3
*:                : SURECSR5
*:                : SURETAS1
*:                : SURETAS2
*:                : SURETAS3
*:                : SURETAS5
*:                : SUREPRO1
*:                : SUREPRO2
*:                : SUREPRO4
*:                : SUREPRO6
*:                : SUREORD1
*:                : SUREORD2
*:                : SUREORD4
*:                : SUREREP1
*:                : SUREREP2
*:                : SUREUTI1
*:                : SUREUTI2
*:                : SUREUTI3
*:                : SUREUTI4
*:                : SUREUTI5
*:                : SUREUTI6
*:                : SUREQUI1
*:                : SUREQUI2
*:                : SUREHEL1
*:                : SUREHEL2
*:
*:       Documented 04/16/94 at 11:01              SNAP!  version 5.00
*:******************************************************************
* instruct.prg
* SURE.prg PROCEDURES
*  the SRC pad's popup and bars
* POPUP src_pop FROM 1,0 MESSAGE "Unit"
* BAR 1 OF src_pop PROMPT "Create Unit from SRC"
*!******************************************************************
*!
*!       Procedure: SURECSR1
*!
*!******************************************************************
PROCEDURE surecsr1
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
   Create Units with fuel and ammo using LINs, personnel strengths, and
   unit designations.  You will use the units you create to form Task
   Organizations, and you will assign Task Organizations to Orders. You
   must create Units to use the Orders-based Reports feature.
ENDTEXT

RETURN
* BAR 2 OF src_pop PROMPT "Edit Unit"
*!******************************************************************
*!
*!       Procedure: SURECSR2
*!
```

228

```
*!************************************************************
PROCEDURE surecsr2
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
    You can change LIN quantities, Unit strengths, Unit Echelons etc. from
    this menu.
ENDTEXT

RETURN
* BAR 3 OF src_pop PROMPT "Create Custom Unit From LINs"
*!************************************************************
*!
*!        Procedure: SURECSR3
*!
*!************************************************************
PROCEDURE surecsr3
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
    If no supplied SRCs are close to the one you need, you may create a
    Custom Unit by choosing LINs and entering the quantity for each LIN.
    You may want to look at a list of LINs available in the system by
    using the System Reports feature from the Main Menu.
ENDTEXT

RETURN
* BAR 5 OF src_pop PROMPT "Delete Unit"
*!************************************************************
*!
*!        Procedure: SURECSR5
*!
*!************************************************************
PROCEDURE surecsr5
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
    DELETE UNITS: You cannot delete units that have been assigned to any
    Task Organization.  You also cannot get them back once deleted, and
    will have to recreate them if you need them again!
ENDTEXT

RETURN
*   the Task pad's popup and bars
* POPUP task_pop FROM 1,5 MESSAGE "Task Organization"
* BAR 1 OF task_pop PROMPT "Create Task Organization from Units"
*!************************************************************
*!
*!        Procedure: SURETAS1
*!
*!************************************************************
PROCEDURE suretas1
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
    You may use the Units you have created and form them into Task
    Organizations.  Task Organizations are groups of units which are
    expected to operate under common battlefield conditions for a
    particular order, sharing the same Parameter Set.  You will assign
    Task Organizations to Orders on which this program will preform its
```

```
         logistics estimations.  You do not have to assign more than one Unit
         to a Task Organization.
ENDTEXT

RETURN
* BAR 2 OF task_pop PROMPT "Create Task Organization by Copying a Previous One"
*!********************************************************************
*!
*!         Procedure: SURETAS2
*!
*!********************************************************************
PROCEDURE suretas2
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
    If you are creating Task Organizations with similar combinations
    of Units, it is often faster to copy a previous one and edit it than
    to start from scratch.
ENDTEXT

RETURN
* BAR 3 OF task_pop PROMPT "Edit Task Organization"
*!********************************************************************
*!
*!         Procedure: SURETAS3
*!
*!********************************************************************
PROCEDURE suretas3
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
    You may assign Profiles, assign and unassign Units etc. from this
    Menu choice.
ENDTEXT

RETURN
* BAR 5 OF task_pop PROMPT "Delete Task Organization"
*!********************************************************************
*!
*!         Procedure: SURETAS5
*!
*!********************************************************************
PROCEDURE suretas5
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
    DELETE TASK ORGANIZATION:  You can only delete Task Organizations
    that have not been assigned to an Order.  You cannot undelete Task
    Organizations once they have been deleted!
ENDTEXT

RETURN
*   the Profiles pad's popup and bars
* POPUP prof_pop FROM 1,14 MESSAGE "Mission Parameter Sets"
* BAR 1 OF prof_pop PROMPT "Create Parameter Sets"
*!********************************************************************
*!
*!         Procedure: SUREPRO1
*!
*!********************************************************************
```

```
PROCEDURE sureprol
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
    Parameter Sets are used to set the conditions for a Task Organizations in
    in particular phase of an Order.  You set parameters for ammunition, fuel,
    medical, food and water consumption in a particular Set, then assign
    that Set later as required.  You must have Parameter Sets assigned to
    Task Organizations within your Orders for logistics calculations to be
    performed using the Orders-based Reports function.
ENDTEXT
RETURN
* BAR 2 OF prof_pop PROMPT "Edit Set"
*!**********************************************************************
*!
*!        Procedure: SUREPRO2
*!
*!**********************************************************************
PROCEDURE surepro2
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
    You can change Parameter settings, including those used for fuel,
    ammunition, food, water and medical supply consumption.  These changes
    affect all Task Organizations in all Orders where the Set has been
    used.
ENDTEXT
RETURN
* BAR 4 OF prof_pop PROMPT "Delete Set"
*!**********************************************************************
*!
*!        Procedure: SUREPRO4
*!
*!**********************************************************************
PROCEDURE surepro4
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
    DELETE MISSION PARAMETER SET:  You can only delete Sets that have not
    been assigned to a Task Organization in an Order.  You cannot undelete
    Sets once they have been deleted!
ENDTEXT
RETURN
* BAR 6 OF prof_pop PROMPT "Custom Area Profiles"
*!**********************************************************************
*!
*!        Procedure: SUREPRO6
*!
*!**********************************************************************
PROCEDURE surepro6
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
    CUSTOM AREA PROFILE:  Non-doctrineal Area Profiles are useful when
    operating conditions affect the number of hours equipment is run or
    kilometers vehicles are expected to go per day.  For example, a
    Custom Profile might be used to specify kilometers for wheeled
    vehicles for a road march.  You may add or delete Custom Profiles,
    but not the doctrinal ones provided with the SURE system.
ENDTEXT
```

```
RETURN
*  the Orders pad's popup and bars
* POPUP orde_pop FROM 1,23 MESSAGE "Orders"
* BAR 1 OF orde_pop PROMPT "Create Order"
*!**********************************************************************
*!
*!       Procedure: SUREORD1
*!
*!*********************************************************************
PROCEDURE sureord1
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
   Orders combine Task Organizations and Profiles and use them with Order
   Phase lengths and strengths to derive consumption calculations.  Each
   Order has a unique Order number, plus an Order name and description.
   You may assign up to nine phases for each Order, each phase of up to
   99.99 days in length, and you may adjust the overall strength of the
   Task Organizations used in each phase.
ENDTEXT
RETURN
* BAR 2 OF orde_pop PROMPT "Edit Order"
*!**********************************************************************
*!
*!       Procedure: SUREORD2
*!
*!*********************************************************************
PROCEDURE sureord2
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
   You may adjust Order phases, phase lengths, Task Organization
   strengths and assign Profiles to Task Organizations for each Phase
   of the Order.
ENDTEXT
RETURN
* BAR 4 OF orde_pop PROMPT "Delete Order"
*!**********************************************************************
*!
*!       Procedure: SUREORD4
*!
*!*********************************************************************
PROCEDURE sureord4
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
   DELETE ORDER:  You cannot undelete Orders once they have been deleted!
ENDTEXT
RETURN
*  the Reports pad's popup and bars
* POPUP repo_pop FROM 1,30 MESSAGE "Reports"
* BAR 1 OF repo_pop PROMPT "Orders-based Reports"
*!**********************************************************************
*!
*!       Procedure: SUREREP1
*!
*!*********************************************************************
PROCEDURE surerep1
@ 9,1 CLEAR TO 16,79
@ 9,1
```

```
TEXT
    Orders-based reports perform logistics estimates on Orders to which
    you have assigned Task Organizations and Parameter Sets.  You can print,
    print to file, or view on screen detailed or consolidated reports for
    ammunition, fuel, or other classes of supply.
ENDTEXT
RETURN
* BAR 4 OF repo_pop PROMPT "System Reports"
*!************************************************************************
*!
*!        Procedure: SUREREP2
*!
*!************************************************************************
PROCEDURE surerep2
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
    System Reports provide information on databases supplied with the SURE
    system, plus those you have created.  They are useful in planning.  For
    example, a list of units you have created could be manually reviewed to
    to plan Task Organizations.
ENDTEXT
RETURN
*   the Utilities pad's popup and bars
* POPUP util_pop FROM 1,38 MESSAGE "Utilities"
* BAR 1 OF util_pop PROMPT "Reindex Files"
*!************************************************************************
*!
*!        Procedure: SUREUTI1
*!
*!************************************************************************
PROCEDURE sureutil
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
    Reindexing recreates all indexes used by SURE. It is not normally needed,
    as SURE's compiled dBase 2.0 system should automatically prompt you if it
    needs to repair an index.  However, if your .MDX index files should become
    deleted and you do not have a current backup, this will recreate the .MDX
    index files. Do NOT use this option without a reason!
ENDTEXT

RETURN
* BAR 2 OF util_pop PROMPT "Backup Files"
*!************************************************************************
*!
*!        Procedure: SUREUTI2
*!
*!************************************************************************
PROCEDURE sureuti2
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
    You may backup your SURE files to a floppy disk.  You must have a blank,
    formatted floppy to start.  If you use a floppy containing a previous SURE
    backup your old files will be overwritten.
ENDTEXT

RETURN
* BAR 3 OF util_pop PROMPT "Restore Files"
```

```
*!********************************************************************
*!
*!        Procedure: SUREUTI3
*!
*!********************************************************************
PROCEDURE sureuti3
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
    If you have a floppy disk with files from a backed-up SURE system you may
    Restore them to this computer.  If you do so, your current files will be
    overwritten, and all data entered in this system will be lost and replaced
    from the files on the floppy disk.  It is HIGHLY recommended that you do
    a backup of this system first before doing a Restore.
ENDTEXT

RETURN
* BAR 4 OF util_pop PROMPT "Printer Setup"
*!********************************************************************
*!
*!        Procedure: SUREUTI4
*!
*!********************************************************************
PROCEDURE sureuti4
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT

    Adjust printer left margin and page length.
ENDTEXT

RETURN
* BAR 5 OF util_pop PROMPT "Turn Compressed Print ON"
*!********************************************************************
*!
*!        Procedure: SUREUTI5
*!
*!********************************************************************
PROCEDURE sureuti5
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
    Many SURE reports must be printed in 17-pitch, compressed print.  If
    your printer does not have a panel switch SURE will try to set the
    option for you.
ENDTEXT

RETURN
* BAR 4 OF util_pop PROMPT "Turn Compressed Print OFF"
*!********************************************************************
*!
*!        Procedure: SUREUTI6
*!
*!********************************************************************
PROCEDURE sureuti6
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT

    If you have set compressed print on from the SURE system, this option will
```

```
     try to restore your printer to normal print.
ENDTEXT

RETURN
*  the Quit pad's popup and bars
* POPUP quit_pop FROM 1,48 MESSAGE "QUIT"
* BAR 1 OF quit_pop PROMPT "Do NOT Quit!"
*!**************************************************************
*!
*!      Procedure: SUREQUI1
*!
*!**************************************************************
PROCEDURE surequi1
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
   Do not quit the SURE system.
ENDTEXT

RETURN
* BAR 2 OF quit_pop PROMPT "END SURE SYSTEM"
*!**************************************************************
*!
*!      Procedure: SUREQUI2
*!
*!**************************************************************
PROCEDURE surequi2
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
   Leave this program and return to the operating system.
ENDTEXT

RETURN
*  the HELP pad's popup and bars
* POPUP help_pop FROM 1,53 MESSAGE "HELP"
* BAR 1 OF help_pop PROMPT "Instructions"
*!**************************************************************
*!
*!      Procedure: SUREHEL1
*!
*!**************************************************************
PROCEDURE surehel1
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
   On-screen display of any part of the SURE Help System.

ENDTEXT

RETURN
* BAR 2 OF help_pop PROMPT "HELP"
*!**************************************************************
*!
*!      Procedure: SUREHEL2
*!
*!**************************************************************
PROCEDURE surehel2
@ 9,1 CLEAR TO 16,79
@ 9,1
```

```
TEXT
   Print the entire SURE Help System.

ENDTEXT

RETURN

*: EOF: INSTRUCT.PRG
```

```
*:***************************************************************
*:
*: Procedure file: D:\SURE\INSTSYSR.PRG
*:
*:  Procs & Fncts: SYSR_BAR1
*:              : SYSR_BAR2
*:              : SYSR_BAR3
*:              : SYSR_BAR4
*:              : SYSR_BAR5
*:              : SYSR_BAR6
*:              : SYSR_BAR7
*:              : SYSR_BAR8
*:              : SYSR_BAR9
*:              : SYSR_BA10
*:              : SYSR_BA11
*:              : SYSR_BA12
*:
*:        Documented 04/16/94 at 11:02            SNAP!  version 5.00
*:***************************************************************
* "SRCs"
*!***************************************************************
*!
*!        Procedure: SYSR_BAR1
*!
*!***************************************************************
PROCEDURE sysr_bar1
@ 16,1 CLEAR TO 20,79
@ 16,1
TEXT
   A list of SRCs from which you may build Units.
ENDTEXT

RETURN
* "SRC with LINs"
*!***************************************************************
*!
*!        Procedure: SYSR_BAR2
*!
*!***************************************************************
PROCEDURE sysr_bar2
@ 16,1 CLEAR TO 20,79
@ 16,1
TEXT
   A list of SRCs with their equipment LINs.  This is a long list, and is best
   printed to a file, then examined with a text editor or word processor.
ENDTEXT

RETURN
* "Units"
*!***************************************************************
*!
*!        Procedure: SYSR_BAR3
*!
*!***************************************************************
PROCEDURE sysr_bar3
@ 16,1 CLEAR TO 20,79
@ 16,1
TEXT
   A list of Unit names.
ENDTEXT
```

```
RETURN
* "Units with LINs"
*!*******************************************************************
*!
*!       Procedure: SYSR_BAR4
*!
*!*******************************************************************
PROCEDURE sysr_bar4
@ 16,1 CLEAR TO 20,79
@ 16,1
TEXT
   A list of Units with their equipment LINs. This is a long list, and is best
   printed to a file, then examined with a text editor or word processor.
ENDTEXT

RETURN
* "Parameter Sets"
*!*******************************************************************
*!
*!       Procedure: SYSR_BAR5
*!
*!*******************************************************************
PROCEDURE sysr_bar5
@ 16,1 CLEAR TO 20,79
@ 16,1
TEXT
   A list of Parameter Sets.
ENDTEXT

RETURN
* "Task Organizations"
*!*******************************************************************
*!
*!       Procedure: SYSR_BAR6
*!
*!*******************************************************************
PROCEDURE sysr_bar6
@ 16,1 CLEAR TO 20,79
@ 16,1
TEXT
   A list of Task Organization names.
ENDTEXT

RETURN
* "Task Organizations with Units"
*!*******************************************************************
*!
*!       Procedure: SYSR_BAR7
*!
*!*******************************************************************
PROCEDURE sysr_bar7
@ 16,1 CLEAR TO 20,79
@ 16,1
TEXT
   A list of Task Organizations and the Units assigned to them.
ENDTEXT

RETURN
* "Orders"
```

```
*!*****************************************************************
*!_
*!        Procedure: SYSR_BAR8
*!
*!*****************************************************************
PROCEDURE sysr_bar8
@ 16,1 CLEAR TO 20,79
@ 16,1
TEXT
   A list of Orders.
ENDTEXT

RETURN
* "Orders with Task Organizations and Parameter Sets"
*!*****************************************************************
*!
*!        Procedure: SYSR_BAR9
*!
*!*****************************************************************
PROCEDURE sysr_bar9
@ 16,1 CLEAR TO 20,79
@ 16,1
TEXT
   A list of Task Organizations with assigned Parameter Sets.
ENDTEXT

RETURN
* "Orders with Task Orgs, Units, and Parameter Sets"
*!*****************************************************************
*!
*!        Procedure: SYSR_BA10
*!
*!*****************************************************************
PROCEDURE sysr_ba10
@ 16,1 CLEAR TO 20,79
@ 16,1
TEXT
   A list of Task Organizations with their assigned Units and Parameter Sets.
   This is may be a long list, and is best printed to a file, then examined
   with a text editor or word processor.
ENDTEXT

RETURN
* "Change Printer Defaults"
*!*****************************************************************
*!
*!        Procedure: SYSR_BA11
*!
*!*****************************************************************
PROCEDURE sysr_ba11
@ 16,1 CLEAR TO 20,79
@ 16,1
TEXT
   Change page length and margin.
ENDTEXT

RETURN
* "Return to Main Menu"
*!*****************************************************************
*!
```

```
*!        Procedure: SYSR_BA12
*!
*!*********************************************************************
PROCEDURE sysr_ba12
@ 16,1 CLEAR TO 20,79
@ 16,1
TEXT
    Return to the Main Menu.
ENDTEXT
RETURN
*: EOF: INSTSYSR.PRG
```

```
*:****************************************************************
*:
*: Procedure file: D:\SURE\INSTTASK.PRG
*:
*:  Procs & Fncts: CRET_BAR1
*:                : CRET_BAR2
*:                : CRET_BAR3
*:                : CRET_BAR4
*:                : CRET_BAR5
*:                : CRET_BAR6
*:                : COPT_BAR1
*:                : COPT_BAR2
*:                : COPT_BAR4
*:                : COPT_BAR3
*:                : EDIT_BAR1
*:                : EDIT_BAR2
*:                : EDIT_BAR3
*:                : EDIT_BAR4
*:                : EDIT_BAR5
*:                : EDIT_BAR6
*:                : EDIT_BAR7
*:                : EDIT_BAR8
*:                : DELT_BAR1
*:                : DELT_BAR2
*:                : DELT_BAR3
*:
*:        Documented 04/16/94 at 11:02              SNAP!  version 5.00
*:****************************************************************
* insttask.prg  Used for Task Organization Menu Items
*!****************************************************************
*!
*!        Procedure: CRET_BAR1
*!
*!****************************************************************
PROCEDURE cret_bar1
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
    Assign a name to your new Task Organization.  Duplicate names are not
    allowed.  After you have assigned a name, you must assign Units, then
    assign Unit quantities, then 'Create from Selected Units' to finish
    creating the Task Organization.

ENDTEXT
RETURN
*!****************************************************************
*!
*!        Procedure: CRET_BAR2
*!
*!****************************************************************
PROCEDURE cret_bar2
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
    Assign Units to your new Task Organization.  You may only assign a Unit
    only once to a Task Organizations, although you may assign a unit to more
    than one Task Organization.  Units are listed in Unit Name order.

ENDTEXT
RETURN
```

```
*!******************************************************************
*!
*!       Procedure: CRET_BAR3
*!
*!******************************************************************
PROCEDURE cret_bar3
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
   Assign Units to your new Task Organization.  You may only assign a Unit
   only once to a Task Organizations, although you may assign a unit to more
   than one Task Organization.  Units are listed in SRC order.

ENDTEXT
RETURN
*!******************************************************************
*!
*!       Procedure: CRET_BAR4
*!
*!******************************************************************
PROCEDURE cret_bar4
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
   You may have up to 99 of this unit in your Task Organization.  You must have
   at least 1.

ENDTEXT
RETURN
*!******************************************************************
*!
*!       Procedure: CRET_BAR5
*!
*!******************************************************************
PROCEDURE cret_bar5
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
   Create your new Task Organization and prepare to create another one.

ENDTEXT
RETURN
*!******************************************************************
*!
*!       Procedure: CRET_BAR6
*!
*!******************************************************************
PROCEDURE cret_bar6
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
   Create your new Task Organization and return to the Main Menu.

ENDTEXT
RETURN
* Create new Task Organization by copying previous one
*!******************************************************************
*!
*!       Procedure: COPT_BAR1
*!
```

```
*!**********************************************************************
PROCEDURE copt_bar1
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
   Assign Units to your new Task Organization by copying from one you have
   previously created.  You may only assign a Unit only once to a Task
   Organizations, although you may assign a unit to more than one Task
   Organization.


ENDTEXT
RETURN
*!**********************************************************************
*!
*!         Procedure: COPT_BAR2
*!
*!**********************************************************************
PROCEDURE copt_bar2
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
   Assign a name to your new Task Organization.  Duplicate names are not
   allowed.




ENDTEXT
RETURN
*!**********************************************************************
*!
*!         Procedure: COPT_BAR4
*!
*!**********************************************************************
PROCEDURE copt_bar4
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
   Create your new Task Organization and return to the Main Menu.

ENDTEXT
RETURN
*!**********************************************************************
*!
*!         Procedure: COPT_BAR3
*!
*!**********************************************************************
PROCEDURE copt_bar3
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
   Create your new Task Organization and prepare to create another one.

ENDTEXT
RETURN
*!**********************************************************************
*!
*!         Procedure: EDIT_BAR1
*!
```

```
*!*******************************************************************
PROCEDURE edit_bar1
@ 12,1 CLEAR TO 16,79
@ 12,1
TEXT
    Select Task Organization to Edit.

ENDTEXT
RETURN
*!*******************************************************************
*!
*!        Procedure: EDIT_BAR2
*!
*!*******************************************************************
PROCEDURE edit_bar2
@ 12,1 CLEAR TO 16,79
@ 12,1
TEXT
    Assign additional Units to the selected Task Organization.  Only
    Units not currently assigned will be shown. Units in Unit Name Order.

ENDTEXT
RETURN
*!*******************************************************************
*!
*!        Procedure: EDIT_BAR3
*!
*!*******************************************************************
PROCEDURE edit_bar3
@ 12,1 CLEAR TO 16,79
@ 12,1
TEXT
    Assign additional Units to the selected Task Organization.  Only
    Units not currently assigned will be shown. Units in SRC Order.

ENDTEXT
RETURN
*!*******************************************************************
*!
*!        Procedure: EDIT_BAR4
*!
*!*******************************************************************
PROCEDURE edit_bar4
@ 12,1 CLEAR TO 16,79
@ 12,1
TEXT
    View the Units in selected Task Organization, Units in Unit Name Order.

ENDTEXT
RETURN
*!*******************************************************************
*!
*!        Procedure: EDIT_BAR5
*!
*!*******************************************************************
PROCEDURE edit_bar5
@ 12,1 CLEAR TO 16,79
@ 12,1
TEXT
    View the Units in selected Task Organization, Units in SRC Order.
```
244

```
ENDTEXT
RETURN
*!*******************************************************************
*!
*!        Procedure: EDIT_BAR6
*!
*!*******************************************************************
PROCEDURE edit_bar6
@ 12,1 CLEAR TO 16,79
@ 12,1
TEXT
   Remove units from the current Task Organization.  This will affect the
   calculations of any Orders to which the current Task Organization is
   assigned. Units in Unit Name Order.

ENDTEXT
RETURN
*!*******************************************************************
*!
*!        Procedure: EDIT_BAR7
*!
*!*******************************************************************
PROCEDURE edit_bar7
@ 12,1 CLEAR TO 16,79
@ 12,1
TEXT
   Remove units from the current Task Organization.  This will affect the
   calculations of any Orders to which the current Task Organization is
   assigned. Units in SRC Order.

ENDTEXT
RETURN
*!*******************************************************************
*!
*!        Procedure: EDIT_BAR8
*!
*!*******************************************************************
PROCEDURE edit_bar8
@ 12,1 CLEAR TO 16,79
@ 12,1
TEXT
   Return to Main Menu.

ENDTEXT
RETURN
* Delete Task Organization
*!*******************************************************************
*!
*!        Procedure: DELT_BAR1
*!
*!*******************************************************************
PROCEDURE delt_bar1
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
   Select Task Organization for deletion.  You will not be able to
   delete any Task Organzations that are currently assigned to an Order.

ENDTEXT
```

245

```
RETURN
*!*******************************************************************
*!
*!        Procedure: DELT_BAR2
*!
*!*******************************************************************
PROCEDURE delt_bar2
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
    Delete Task Organization that you have selected.  You will not be able
    to get deleted Task Organizations back!

ENDTEXT
RETURN
*!*******************************************************************
*!
*!        Procedure: DELT_BAR3
*!
*!*******************************************************************
PROCEDURE delt_bar3
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
    Return to Main Menu.

ENDTEXT
RETURN

*: EOF: INSTTASK.PRG
```

```
*:******************************************************************
*:
*:  Procedure file: D:\SURE\INSTUNIT.PRG
*:
*:   Procs & Fncts: CSRC_BAR1
*:                : CSRC_BAR2
*:                : CSRC_BAR4
*:                : CSRC_BAR5
*:                : CSRC_BAR6
*:                : CSRC_BAR3
*:                : EDIU_BAR1
*:                : EDIU_BAR2
*:                : EDIU_BAR3
*:                : EDIU_BAR4
*:                : EDIU_BAR5
*:                : EDIU_BAR6
*:                : EDIU_BAR7
*:                : EDIU_BAR8
*:                : CUST_BAR1
*:                : CUST_BAR2
*:                : CUST_BAR3
*:                : CUST_BAR4
*:                : CUST_BAR5
*:                : CUST_BAR6
*:                : DELU_BAR1
*:                : DELU_BAR2
*:                : DELU_BAR3
*:                : DELU_BAR4
*:
*:       Documented 04/16/94 at 11:02            SNAP!  version 5.00
*:******************************************************************
* instunit.prg Procedure files for items in Unit Menu
* CREATESRC.prg PROCEDURES
*!******************************************************************
*!
*!       Procedure: CSRC_BAR1
*!
*!******************************************************************
PROCEDURE csrc_bar1
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
   Provided SRC's include unit descriptions, unit strengths, and
   LIN information for each TOE item that consumes fuel or ammunition.
   You should select an SRC to base each unit in your organization
   on.  You may modify LINs and unit strengths from the Edit Unit program,
   but unit names, once assigned, are permanent and cannot be duplicated.
   It is recommended that you do NOT use the standard SRC description as
   a unit name, but modify it to meet your needs.
ENDTEXT
RETURN
*!******************************************************************
*!
*!       Procedure: CSRC_BAR2
*!
*!******************************************************************
PROCEDURE csrc_bar2
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
```

247

```
    If you have a list of your organization's units with their SRC codes,
    it is substantially faster to let this program search for them by      -
    number rather than scrolling through the list using the Browse feature.
    You do not have to enter a complete number, even a partial one will get
    you to about the right place to do a manual final scroll.
ENDTEXT
RETURN
*!*********************************************************************
*!
*!        Procedure: CSRC_BAR4
*!
*!*********************************************************************
PROCEDURE csrc_bar4
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
    If you have already created at least one unit from an SRC, you
    may copy its existing data, complete with LIN and strength information,
    to use with another unit.  This is a valuable time saver when doing
    several units that are essentially similar.
ENDTEXT
RETURN
*!*********************************************************************
*!
*!        Procedure: CSRC_BAR5
*!
*!*********************************************************************
PROCEDURE csrc_bar5
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
    You may view the SRC and Unit Name of units you have already created.
    This feature will help you avoid trying to duplicate unit names.  If
    you need a complete listing of unit with all data, you may want to use
    the Reports feature from the Main Menu.
ENDTEXT
RETURN
*!*********************************************************************
*!
*!        Procedure: CSRC_BAR6
*!
*!*********************************************************************
PROCEDURE csrc_bar6
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
    If you are done creating units, press Enter now to return
    to the Main Menu.
ENDTEXT
RETURN
*!*********************************************************************
*!
*!        Procedure: CSRC_BAR3
*!
*!*********************************************************************
PROCEDURE csrc_bar3
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
    If you wish to add this unit to your unit list, and then create another
```

```
     unit from an SRC, press Enter now.
ENDTEXT
RETURN
* Procedures for editunit.prg
*!***********************************************************************
*!
*!       Procedure: EDIU_BAR1
*!
*!***********************************************************************
PROCEDURE ediu_bar1
@ 11,1 CLEAR TO 16,79
@ 11,1
TEXT
   Select a Unit to view or make changes to.  This option allow you
   to pick from your Unit list in Unit Name order.
ENDTEXT
RETURN
*!***********************************************************************
*!
*!       Procedure: EDIU_BAR2
*!
*!***********************************************************************
PROCEDURE ediu_bar2
@ 11,1 CLEAR TO 16,79
@ 11,1
TEXT
   Select a Unit to view or make changes to. This option allows you to
   optionally enter an SRC, then pick from your Unit list in SRC order.
ENDTEXT
RETURN
*!***********************************************************************
*!
*!       Procedure: EDIU_BAR3
*!
*!***********************************************************************
PROCEDURE ediu_bar3
@ 11,1 CLEAR TO 16,79
@ 11,1
TEXT
   Add or subtract to quantities of equipment assigned to selected unit.
ENDTEXT
RETURN
*!***********************************************************************
*!
*!       Procedure: EDIU_BAR4
*!
*!***********************************************************************
PROCEDURE ediu_bar4
@ 11,1 CLEAR TO 16,79
@ 11,1
TEXT
   Add a new item of equipment to selected unit.  If it is already assigned,
   a duplicate will not be added.
ENDTEXT
RETURN
*!***********************************************************************
*!
*!       Procedure: EDIU_BAR5
*!
*!***********************************************************************
```

```
PROCEDURE ediu_bar5
@ 11,1 CLEAR TO 16,79
@ 11,1
TEXT
    Delete an item of equipment from selected unit.
ENDTEXT
RETURN
*!*****************************************************************
*!
*!        Procedure: EDIU_BAR6
*!
*!*****************************************************************
PROCEDURE ediu_bar6
@ 11,1 CLEAR TO 16,79
@ 11,1
TEXT
    Change number of personnel assigned to selected unit.
ENDTEXT
RETURN
*!*****************************************************************
*!
*!        Procedure: EDIU_BAR7
*!
*!*****************************************************************
PROCEDURE ediu_bar7
@ 11,1 CLEAR TO 16,79
@ 11,1
TEXT
    Change unit echelon.  This entry is used for water comsumption calculations.
ENDTEXT
RETURN
*!*****************************************************************
*!
*!        Procedure: EDIU_BAR8
*!
*!*****************************************************************
PROCEDURE ediu_bar8
@ 11,1 CLEAR TO 16,79
@ 11,1
TEXT
    Return to Main Menu.
ENDTEXT
RETURN
* Custom Unit
*!*****************************************************************
*!
*!        Procedure: CUST_BAR1
*!
*!*****************************************************************
PROCEDURE cust_bar1
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
    You may initiate a new, custom unit and assign it a permanent unit name. No
    LIN items of equipment will be automatically added, but you may choose as
    many as you wish from the list of LINs and assign quantities.
ENDTEXT
RETURN
*!*****************************************************************
*!
```

```
*!     ,     Procedure: CUST_BAR2
*!
*!***************************************************************
PROCEDURE cust_bar2
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
   Examine equipment you have assigned to this unit and optionally change
   quantities.
ENDTEXT
RETURN
*!***************************************************************
*!
*!        Procedure: CUST_BAR3
*!
*!***************************************************************
PROCEDURE cust_bar3
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
   Add additional equipment LINs.
ENDTEXT
RETURN
*!***************************************************************
*!
*!        Procedure: CUST_BAR4
*!
*!***************************************************************
PROCEDURE cust_bar4
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
   Delete equipment LINs.
ENDTEXT
RETURN
*!***************************************************************
*!
*!        Procedure: CUST_BAR5
*!
*!***************************************************************
PROCEDURE cust_bar5
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
   This program tries to guess the echelon of the unit from the name that
   you entered.  You may change it here if necessary.  Echelon data is used
   in calculating water consumption.
ENDTEXT
RETURN
*!***************************************************************
*!
*!        Procedure: CUST_BAR6
*!
*!***************************************************************
PROCEDURE cust_bar6
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
   Creates unit from data you have choosen and returns to Main Menu.
ENDTEXT
```

```
RETURN
* Delete Unit
*!********************************************************************
*!
*!        Procedure: DELU_BAR1
*!
*!********************************************************************

PROCEDURE delu_bar1
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
    Select Unit for deletion.  You will not be able to delete any units that
    are currently assigned to a Task Organization. Units are in Unit Name
    Order.
ENDTEXT
RETURN
* Delete Unit
*!********************************************************************
*!
*!        Procedure: DELU_BAR2
*!
*!********************************************************************
PROCEDURE delu_bar2
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
    Select Unit for deletion.  You will not be able to delete any units that
    are currently assigned to a Task Organization. Units are in Unit Order.
ENDTEXT
RETURN
* Delete Unit
*!********************************************************************
*!
*!        Procedure: DELU_BAR3
*!
*!********************************************************************
PROCEDURE delu_bar3
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
    Delete Unit that you have selected.  You will not be able to get deleted
    Unit back!
ENDTEXT
RETURN
* Delete Unit
*!********************************************************************
*!
*!        Procedure: DELU_BAR4
*!
*!********************************************************************
PROCEDURE delu_bar4
@ 10,1 CLEAR TO 16,79
@ 10,1
TEXT
    Return to Main Menu.
ENDTEXT
RETURN

*: EOF: INSTUNIT.PRG
```

```
*:*******************************************************************
*:
*: Procedure file: D:\SURE\MEMOPAGR.PRG
*:
*:  Procs & Fncts: MEMOPAGR()
*:
*:       Documented 04/16/94 at 11:02              SNAP!  version 5.00
*:*******************************************************************
*!*******************************************************************
*!
*!         Function: MEMOPAGR()
*!
*!        Called by: HELP.PRG
*!                 : HELPNDX.PRG
*!                 : EDITHELP.PRG
*!
*!*******************************************************************
FUNCTION memopagr
*------------------------------------------------------------------
-
*-- Programmer..: Martin Leon (HMAN - ATBBS/Borland BBS) Modified by John
*-- Friedson
*-- Date........: 10/28/1991
*-- Notes.......: Used to display a memo on screen, allowing user to scroll
*--               memo at will.
*-- Written for.: dBASE IV, 1.1
*-- Rev. History: 10/28/1991 -- Original  03/01/94 Modifications
*-- Calls.......: None
*-- Called by...: Any
*-- Usage.......: ?MemoPagr(<cMemo>,<ULRow>,<ULCol>,<BRRow>,<BRCol>)
*-- Example.....: ?MemoPagr(MoreData,10,20,20,65)
*-- Returns.....: .F.
*-- Parameters..: cMemo    = name of memo field
*--               nULRow   = upper left row position
*--               nULCol   = upper left column position
*--               nBRRow   = bottom right row position
*--               nBRCol   = bottom right column position
*------------------------------------------------------------------
-
PARAMETER cmemo, nulrow, nulcol, nbrrow, nbrcol
PRIVATE ccursor, nesc, npgdn, npgup, nup, ndn, nnumlines,nlines,nkey
PRIVATE natline,natrow
cmemo="help->notes"
*-- set environment
SET MEMOWIDTH TO nbrcol - nulcol - 1
ccursor = SET( "CURSOR" )
SET CURSOR OFF
*-- define a few keys
nesc  = 27
npgdn = 3
npgup = 18
nup   = 5
ndn   = 24
nenter= 13
*-- determine size of window
nnumlines = MEMLINES(&cmemo)
nlines = nbrrow - nulrow - 1
*-- save the screen, so we can restore it
SAVE SCREEN TO stmp
@ nulrow+1, nulcol+1 CLEAR TO nbrrow+1, nbrcol+1
```

```
@ nulrow+1, nulcol+1 FILL TO nbrrow+1, nbrcol+1 COLOR B/N
@ nulrow+1, nulcol+1 FILL TO nbrrow-1, nbrcol-1 COLOR RG+/B
@ nulrow, nulcol TO nbrrow, nbrcol DOUBLE COLOR RG+/B
*-- deal with a blank memo ...
IF nnumlines = 0
    @ nulrow + 1, nulcol + 1 SAY ;
        "Blank Memo.  Press any key to continue..." COLOR RG+/B
    nkey = INKEY(0)
    *-- reset the whole thing
    RESTORE SCREEN FROM stmp
    RELEASE SCREEN stmp
    SET CURSOR &ccursor
    RETURN .F.
ENDIF (nnumlines = 0)
natline = 1
natrow = 1
DO WHILE natline <= nnumlines
    *-- Show one window full
    DO WHILE natrow <= nlines .AND. natline <= nnumlines
        @ nulrow+natrow, nulcol + 1 SAY ;
            MLINE( &cmemo, natline ) COLOR RG+/B
        natline = natline + 1
        natrow = natrow + 1
    ENDDO (natrow <= nlines .AND. natline <= nnumlines)
    *-- If at last line of memo...
    IF natline > nnumlines
        *-- If memo is shorter than one page, put box character in
        *-- bottom left corner of box, otherwise, put an up arrow
        *-- symbol there.
        @ nbrrow - 1, nbrcol SAY ;
            IIF( nnumlines <= nlines, CHR(186), CHR(24)) COLOR W+/B
        DO WHILE .T.
            nkey = INKEY(0)
            *-- If memo is shorter than one page, only allow Esc key
            IF nnumlines <= nlines
                IF nkey = nesc .OR. nkey = nenter
                    EXIT
                ENDIF (nkey = nesc .OR. nkey = nenter)
                *-- Otherwise, allow Esc or PgUp keys
            ELSE
                IF nkey = nesc .OR. nkey = npgup .OR. nkey = nup
                    EXIT
                ENDIF (nkey = nesc .OR. nkey = npgup .OR. nkey = nup)
            ENDIF (nnumlines <= nlines)
            ?? CHR(7)
        ENDDO (.T.)
        IF nkey = nesc .OR. nkey = nenter
            RESTORE SCREEN FROM stmp
            RELEASE SCREEN stmp
            SET CURSOR &ccursor
            RETURN .F.
        ENDIF (nkey = nesc .OR. nkey = nenter)
        @ nulrow+1, nulcol+1 CLEAR TO nbrrow-1, nbrcol-1
        @ nulrow+1, nulcol+1 FILL TO nbrrow-1, nbrcol-1 ;
            COLOR RG+/B
        natline = natline -  natrow - nlines + 1
        natline = IIF( natline < 1, 1, natline )
        natrow = 1
        LOOP
    ENDIF (natline > nnumlines)
```

```
*-- Not at end of memo yet...
*-- If on first page, show down arrow only, otherwise show
*-- up/down arrow on border of box.
@ nbrrow - 1, nbrcol SAY ;
    IIF( natline - nlines = 1, CHR(25), CHR(18)) COLOR W+/B
DO WHILE .T.
    nkey = INKEY(0)
    *-- If this is the first page of the memo on screen...
    IF natline - nlines = 1
        *-- Only honor PgDn, up cursor, and Esc keys
        IF nkey = npgdn .OR. nkey = ndn .OR. nkey = nesc .OR. nkey = nenter
            EXIT
        ENDIF (nkey = npgdn .OR. nkey = ndn .OR. nkey = nesc .OR. nkey =
nenter)
        *-- otherwise honor PgUp and up cursor as well key as well
    ELSE
        IF nkey = npgup .OR. nkey = nup .OR. nkey = npgdn .OR. ;
            nkey = ndn .OR. nkey = nesc .OR. nkey = nenter
            EXIT
        ENDIF (nkey = npgup .OR. nkey = nup .OR. nkey = npgdn .OR. ;)
    ENDIF (natline - nlines = 1)
    ?? CHR(7)
ENDDO (.T.)
DO CASE
CASE nkey = nesc .OR. nkey = nenter
    RESTORE SCREEN FROM stmp
    RELEASE SCREEN stmp
    SET CURSOR &ccursor
    RETURN .F.
CASE nkey = npgup .OR. nkey = nup
    @ nulrow+1, nulcol+1 CLEAR TO nbrrow-1, nbrcol-1
    @ nulrow+1, nulcol+1 FILL TO nbrrow-1, nbrcol-1 ;
        COLOR RG+/B
    natline = (natline - (2 * nlines))
    natline = IIF( natline < 1, 1, natline )
    natrow = 1
    LOOP
CASE nkey = npgdn .OR. nkey = ndn
    @ nulrow+1, nulcol+1 CLEAR TO nbrrow-1, nbrcol-1
    @ nulrow+1, nulcol+1 FILL TO nbrrow-1, nbrcol-1 ;
        COLOR RG+/B
    natrow = 1
    LOOP
ENDCASE
ENDDO (natline <= nnumlines)
RETURN .F.
*-- EoF: MemoPagr()

*: EOF: MEMOPAGR.PRG
```

```
*:*********************************************************************
*:
*:          Program: D:\SURE\MEMVARS.PRG
*:
*:          Called by: SURE.PRG
*:
*:          Documented 04/16/94 at 11:01                SNAP!  version 5.00
*:*********************************************************************
* SURE Memory variable initialization.
* Primary use is for public variables.
* Variable created by dBase screen generators, etc. are not listed here.
RETURN
*: EOF: MEMVARS.PRG
```

```
*:**********************************************************************
*:
*: Procedure file: D:\SURE\ORDEREP1.PRG
*:
*:   Procs & Fncts: ORDR_PRO
*:               : OLDORD
*:               : DISPORD3
*:               : STARTORD
*:               : ORDREND
*:
*:         Set by: REPO_PRO               (procedure in SURE.PRG)
*:
*:          Calls: ORDR_PRO               (procedure in ORDEREP1.PRG)
*:
*:         Documented 04/16/94 at 11:01              SNAP!  version 5.00
*:**********************************************************************
* orderrep.prg Orders-based logistics estimations
SET PROCEDURE TO instrept
* PRIVATE Variables
STORE PROGRAM() TO progvar
**********************************
STORE 1 TO statusva
STORE .n. TO status1v
STORE .n. TO status2v
STORE .n. TO status3v
STORE .n. TO status4v
STORE 0 TO polvar
STORE 0 TO ammovar
STORE 0 TO linevar
STORE 0 TO internal
STORE 0 TO calcvar
STORE 0 TO phaselnv
STORE 0 TO numberva
STORE SPACE(60) TO taskorgvar
STORE SPACE(8) TO filevar
STORE SPACE(9) TO phasevar
STORE .n. TO parentye
STORE "TXT" TO extvar
STORE "NO " TO yesnovar
STORE SPACE(12) TO fileext
STORE SPACE(60) TO namevar
CLEAR
DEFINE POPUP ordr_pop FROM 1,0
DEFINE BAR 1 OF ordr_pop PROMPT "Select Order/Calcuate Consumption"
DEFINE BAR 2 OF ordr_pop PROMPT "POL Estimate"
DEFINE BAR 3 OF ordr_pop PROMPT "Ammo Estimate"
DEFINE BAR 4 OF ordr_pop PROMPT "Other Supply Estimates"
DEFINE BAR 5 OF ordr_pop PROMPT "Return to Main Menu"
ON SELECTION POPUP ordr_pop DO ordr_pro
ON BAR 1 OF ordr_pop DO ordr_bar1
ON BAR 2 OF ordr_pop DO ordr_bar2
ON BAR 3 OF ordr_pop DO ordr_bar3
ON BAR 4 OF ordr_pop DO ordr_bar4
ON BAR 5 OF ordr_pop DO ordr_bar5
DO WHILE PROMPT()<>"Return to Main Menu"
   ACTIVATE POPUP ordr_pop
ENDDO (PROMPT()<>"Return to Main Menu")
*!**********************************************************************
*!
*!       Procedure: ORDR_PRO
```

```
*!
*!        Called by: ORDEREP1.PRG
*!
*!            Calls: STARTORD          (procedure in ORDEREP1.PRG)
*!                 : OLDORD            (procedure in ORDEREP1.PRG)
*!                 : GETORDER          (procedure in EDITORDE.PRG)
*!                 : DISPORD3          (procedure in ORDEREP1.PRG)
*!                 : PRINTPOL.PRG
*!                 : PRINTAMM.PRG
*!                 : PRINTOTH.PRG
*!                 : ORDREND           (procedure in ORDEREP1.PRG)
*!
*!**********************************************************************
PROCEDURE ordr_pro
DO CASE
    * Select order
CASE BAR() = 1
    IF statusva<>1
        CLOSE DATA
        STORE 1 TO statusva
        STORE .n. TO status1v
        STORE .n. TO status2v
        STORE .n. TO status3v
        STORE .n. TO status4v
        STORE 0 TO polvar
        STORE 0 TO ammovar
        STORE 0 TO linevar
        STORE 0 TO internal
        STORE 0 TO calcvar
        STORE SPACE(60) TO taskorgvar
        STORE SPACE(8) TO filevar
        STORE SPACE(9) TO phasevar
        STORE .n. TO parentye
        STORE "TXT" TO extvar
        STORE SPACE(12) TO fileext
    ENDIF (statusva<>1)
    DO startord
    * Make sure there are some records to view!
    IF RECCOUNT()=0
        ACTIVATE WINDOW small
        ? CHR(7)
        @ 1,2 SAY "You must have at"
        @ 2,2 SAY "least one Order!"
        @ 3,3 SAY "Press any key..."
        WAIT ""
        DEACTIVATE WINDOW small
        CLEAR
        RETURN
    ENDIF (RECCOUNT()=0)
    DO oldord
    IF statusva<>2
        DO getorder
    ELSE
        DO dispord3
    ENDIF (statusva<>2)
    statusva=2
    * POL Estimate
CASE BAR() = 2
    IF statusva=1
        ? CHR(7)
```

```
          ACTIVATE WINDOW small
          @ 1,1 SAY "Choose order first!"
          @ 3,1
          WAIT
          DEACTIVATE WINDOW small
          RETURN
      ENDIF (statusva=1)
      DO printpol
      * Ammo Estimate
   CASE BAR() = 3
      IF statusva=1
          ? CHR(7)
          ACTIVATE WINDOW small
          @ 1,1 SAY "Choose order first!"
          @ 3,1
          WAIT
          DEACTIVATE WINDOW small
          RETURN
      ENDIF (statusva=1)
      DO printamm
      * Other Supply Estimates
   CASE BAR() = 4
      IF statusva=1
          ? CHR(7)
          ACTIVATE WINDOW small
          @ 1,1 SAY "Choose order first!"
          @ 3,1
          WAIT
          DEACTIVATE WINDOW small
          RETURN
      ENDIF (statusva=1)
      DO printoth
      * DONE Save SRC and end
   CASE BAR() = 5
      DO ordrend
   ENDCASE
   * Give the user a chance to use the last estimate without recalculating
   *!*****************************************************************
   *!
   *!       Procedure: OLDORD
   *!
   *!       Called by: ORDR_PRO            (procedure in ORDEREP1.PRG)
   *!
   *!*****************************************************************
   PROCEDURE oldord
   IF RECCOUNT("cl_other")>0
      IF SEEK(cl_other->number)
          ACTIVATE WINDOW process
          TEXT
      Do you want to use the last set of caclulations done for the order
      listed below?  If you do, you will save time, but any changes you
      may have made to Units, Task Organizations, Parameter Sets or to the
      Order itself will NOT be taken into effect!

          ENDTEXT
          @ ROW(), COL() SAY "Last Order Number: "+orderlst->number
          ?
          @ ROW(), COL() SAY "               Name: "+orderlst->name
          ?
          @ ROW(), COL() SAY "         Description: "+orderlst->descriptio
```

259

```
         ?                               ,
         @ ROW(), COL() SAY "Enter YES if you want to use this Orders old data! "
         @ ROW(), COL() GET yesnovar PICTURE "!!!" VALID yesnovar="NO" .OR.
   yesnovar="YES";
             ERROR "Must be YES or NO"
         READ

         IF yesnovar="YES"
             statusva=2
         ENDIF (yesnovar="YES")
         DEACTIVATE WINDOW process
     ENDIF (SEEK(cl_other->number))
ENDIF (RECCOUNT("cl_other")>0)
*!***************************************************************************
*!
*!       Procedure: GETORDER
*!
*!       Called by: ORDT_PRO            (procedure in EDITORDE.PRG)
*!                : DORD_PRO            (procedure in DELORDE.PRG)
*!                : ORDR_PRO            (procedure in ORDEREP1.PRG)
*!
*!          Calls: SETBROWS.PRG
*!                : ENDBROWS.PRG
*!                : DISPORD2            (procedure in CREATEOR.PRG)
*!                : VIEWORDE            (procedure in CREATEOR.PRG)
*!                : DISPORD             (procedure in CREATEOR.PRG)
*!                : DISPORD3            (procedure in ORDEREP1.PRG)
*!                : CALCPOL.PRG
*!                : CALCAMMO.PRG
*!                : CALCOTHE.PRG
*!
*!***************************************************************************
PROCEDURE getorder
DO vieworde
STORE NUMBER TO numberva
SET MESSAGE TO "Working..."
statusva=2
* Show orderlst info on screen
DO dispord3
* Advise user to wait
CLEAR
* Prepare estimation database
ACTIVATE WINDOW process
ACTIVATE SCREEN
@ 16,2 SAY "Please Wait, doing POL  "
ACTIVATE WINDOW process
DO calcpol
ACTIVATE SCREEN
@ 16,2 SAY "Please Wait, doing AMMO "
ACTIVATE WINDOW process
DO calcammo
ACTIVATE SCREEN
@ 16,2 SAY "Please Wait, doing OTHER "
ACTIVATE WINDOW process
DO calcothe
DEACTIVATE WINDOW process
* Let user know calculations are done
? CHR(7)
SET MESSAGE TO
CLEAR
*****************************************************************
```

```
*!***********************************************************************
*!
*!          Procedure: VIEWORDE
*!
*!          Called by: ORDS_PRO              (procedure in CREATEOR.PRG)
*!                    : ORDT_PRO              (procedure in EDITORDE.PRG)
*!                    : GETORDER              (procedure in EDITORDE.PRG)
*!
*!             Calls: SETBROWS.PRG
*!                  : ENDBROWS.PRG
*!                  : DISPORD2               (procedure in CREATEOR.PRG)
*!                  : DISPORD3               (procedure in ORDEREP1.PRG)
*!
*!***********************************************************************
PROCEDURE vieworde
SELECT 3
* Make sure there are some records to view
IF RECCOUNT()=0
   ACTIVATE WINDOW small
   ? CHR(7)
   @ 1,2 SAY "You must have at"
   @ 2,2 SAY "least one Order!"
   @ 3,3 SAY "Press any key..."
   WAIT ""
   DEACTIVATE WINDOW small
   CLEAR
   RETURN
ENDIF (RECCOUNT()=0)
ACTIVATE WINDOW browser
GO TOP
*      SET MESSAGE TO "Choose an Order to View"
DO setbrows
BROWSE NOMENU NOEDIT NOAPPEND NODELETE FIELDS NUMBER,name, descriptio FREEZE
name COMPRESS
DO endbrows
DEACTIVATE WINDOW browser
SET CURSOR OFF
CLEAR
DO WHILE LASTKEY()<>23
   DO dispord3
   IF RECCOUNT()=1
      @ 23,30 SAY "Crtl-W when Done"
   ELSE
      @ 23,15 SAY "Page-Up or Page-Down to View, Ctrl-W when Done"
   ENDIF (RECCOUNT()=1)
   IF LASTKEY()=3
      SKIP
      IF EOF()
         GO TOP
      ENDIF (EOF())
   ENDIF (LASTKEY()=3)
   IF LASTKEY()=18
      SKIP -1
      IF BOF()
         GO BOTTOM
      ENDIF (BOF())
   ENDIF (LASTKEY()=18)
   WAIT ""
ENDDO (LASTKEY()<>23)
SET CURSOR ON
```

261

```
SET MESSAGE TO
*!*******************************************************************
*!
*!        Procedure: DISPORD3
*!
*!        Called by: VIEWORDE         (procedure in CREATEOR.PRG)
*!                 : GETORDER          (procedure in EDITORDE.PRG)
*!                 : ORDR_PRO          (procedure in ORDEREP1.PRG)
*!
*!*******************************************************************
PROCEDURE dispord3
@ 20,2 SAY "Order Number:   "
@ ROW(),COL() SAY NUMBER
@ 21,2 SAY "  Order Name:   "
@ ROW(),COL() SAY name
@ 22,2 SAY "Description:   "
@ ROW(), COL() SAY descriptio
@ 23,2 SAY "Number of Phases:   "
@ ROW(), COL() SAY LTRIM(STR(phases))
*!*******************************************************************
*!
*!        Procedure: STARTORD
*!
*!        Called by: ORDR_PRO          (procedure in ORDEREP1.PRG)
*!
*!             Uses: CL_OTHER.DBF
*!                 : ORDERLST.DBF
*!
*!        MDX files: CL_OTHER.MDX
*!                 : ORDERLST.MDX
*!
*!*******************************************************************
PROCEDURE startord
* OPEN FILES
CLOSE DATA
STORE PROGRAM() TO progvar
SELECT 26
USE cl_other
SELECT 3
USE orderlst ORDER NUMBER
*!*******************************************************************
*!
*!        Procedure: ORDREND
*!
*!        Called by: ORDR_PRO          (procedure in ORDEREP1.PRG)
*!
*!*******************************************************************
PROCEDURE ordrend
CLOSE DATA
DEACTIVATE POPUP
RETURN

*: EOF: ORDEREP1.PRG
```

```
*:*********************************************************************
*:
*: Procedure file: D:\SURE\PRINTAMM.PRG
*:
*:  Procs & Fncts: AMMOCALC
*:              : PUNITAMM
*:              : PTASKAMM
*:              : PORDEAMM
*:              : LINAMM
*:
*:        Set by: ORDR_PRO          (procedure in ORDEREP1.PRG)
*:
*:        Calls: STARTPLR           (procedure in PRINTPOL.PRG)
*:             : DETA_PRO           (procedure in PRINTPOL.PRG)
*:             : PRIN_PRO           (procedure in SYSREP.PRG)
*:
*:        Documented 04/16/94 at 11:02              SNAP!  version 5.00
*:*********************************************************************
STORE PROGRAM() TO progvar
STORE SPACE(39) TO nomenvar
STORE SPACE(60) TO filevar
STORE " NO" TO detailva
STORE " " TO printdes
STORE "    " TO vttlva
STORE SPACE(9) TO plenvar
STORE SPACE(10) TO amm1var
STORE SPACE(9) TO amm2var
* Store Order Weight & Cube Totals
STORE 0 TO attlwgt
STORE 0 TO attlcube
* Store Order Weight & Cube per Phase
STORE 0 TO pttlwgt
STORE 0 TO pttlcube
* STORE Unit Phase Weight & Cube Totals per Phase
STORE 0 TO ottlwgt
STORE 0 TO ottlcube
* Store Task Force Weight & cube Totals per Phase
STORE 0 TO tttlwgt
STORE 0 TO tttlcube
STORE 1 TO qtycount
STORE 1 TO unitqty
CLEAR
* Setup Printers for Condensed Print
* Is printer on line?
IF PRINTSTATUS()
   DO CASE
   CASE vprintva="Dot Matrix"
      ??? CHR(27)+CHR(15)
   CASE vprintva="Laser"
   CASE vprintva="DeskJet"
      ??? CHR(27)+"(s16.66H"
   ENDCASE
ELSE
   ACTIVATE WINDOW small
   @ 1,1 SAY "Printer not ready,"
   @ 2,2 SAY "compressed print NOT set!"
   @ 3,3
   WAIT
   DEACTIVATE WINDOW small
ENDIF (PRINTSTATUS())
```

```
DO startplr
DEFINE POPUP deta_pop FROM 1,1
DEFINE BAR 1 OF deta_pop PROMPT "LIN by LIN Detail, plus all Totals"
DEFINE BAR 2 OF deta_pop PROMPT "Summary Report - Unit, Task Org, and Order
Totals"
DEFINE BAR 3 OF deta_pop PROMPT "Summary Report - Task Org and Order Totals
Only"
DEFINE BAR 4 OF deta_pop PROMPT "Summary Report - Order Totals Only"
ON SELECTION POPUP deta_pop DO deta_pro
ON BAR 1 OF deta_pop DO deta_bar1
ON BAR 2 OF deta_pop DO deta_bar2
ON BAR 3 OF deta_pop DO deta_bar3
ON BAR 4 OF deta_pop DO deta_bar4
DEFINE POPUP prin_pop FROM 1,1
DEFINE BAR 1 OF prin_pop PROMPT "Print Estimate to Printer"
DEFINE BAR 2 OF prin_pop PROMPT "Print Estimate to Screen"
DEFINE BAR 3 OF prin_pop PROMPT "Print Estimate to ASCII File"
DEFINE BAR 4 OF prin_pop PROMPT "Change Printer Defaults"
DEFINE BAR 5 OF prin_pop PROMPT "Return to Previous Menu"
ON SELECTION POPUP prin_pop DO prin_pro
ON BAR 1 OF prin_pop DO prin_bar1
ON BAR 2 OF prin_pop DO prin_bar2
ON BAR 3 OF prin_pop DO prin_bar3
ON BAR 4 OF prin_pop DO prin_bar4
ON BAR 5 OF prin_pop DO prin_bar5
DO WHILE PROMPT()<>"Return to Previous Menu"
   ACTIVATE POPUP prin_pop
ENDDO (PROMPT()<>"Return to Previous Menu")
*!********************************************************************
*!
*!         Procedure: PRIN_PRO
*!
*!         Called by: SYSREP.PRG
*!                  : PRINTPOL.PRG
*!                  : PRINTAMM.PRG
*!                  : PRINTOTH.PRG
*!
*!            Calls: PRNSETUP.PRG
*!                 : STARTPLR              (procedure in PRINTPOL.PRG)
*!                 : REPORTHD              (procedure in PRINTPOL.PRG)
*!                 : TFHEAD                (procedure in PRINTPOL.PRG)
*!                 : LINPOL                (procedure in PRINTPOL.PRG)
*!                 : POLCALC               (procedure in PRINTPOL.PRG)
*!                 : PUNITPOL              (procedure in PRINTPOL.PRG)
*!                 : PTASKPOL              (procedure in PRINTPOL.PRG)
*!                 : PORDEPOL              (procedure in PRINTPOL.PRG)
*!                 : LINAMM                (procedure in PRINTAMM.PRG)
*!                 : AMMOCALC              (procedure in PRINTAMM.PRG)
*!                 : PUNITAMM              (procedure in PRINTAMM.PRG)
*!                 : PTASKAMM              (procedure in PRINTAMM.PRG)
*!                 : PORDEAMM              (procedure in PRINTAMM.PRG)
*!                 : OTHECALC              (procedure in PRINTOTH.PRG)
*!                 : PUNITOTH              (procedure in PRINTOTH.PRG)
*!                 : PTASKOTH              (procedure in PRINTOTH.PRG)
*!                 : PORDEOTH              (procedure in PRINTOTH.PRG)
*!
*!********************************************************************
PROCEDURE prin_pro
DO CASE
CASE BAR()=1
```

```
        printdes="p"
CASE BAR()=2
        printdes="s"
        CLEAR
CASE BAR()=3
        printdes="f"
        ACTIVATE WINDOW process
        @ 1,1 SAY "Enter path and file name."
        @ 3,1 GET filevar
        READ
        *filevar=LTRIM(TRIM(filevar))
        SET ALTERNATE TO &filevar
        DEACTIVATE WINDOW process
CASE BAR()=4
        DO prnsetup
CASE BAR()=5
        DEACTIVATE POPUP
        RETURN
ENDCASE
CLEAR
ACTIVATE POPUP deta_pop
* Set according to destination
DO CASE
CASE printdes="p"
        ACTIVATE WINDOW process
        SET PRINT ON
        SET DEVICE TO PRINT
CASE printdes="s"
CASE printdes="f"
        SET ALTERNATE ON
ENDCASE
* Initialize Variables
STORE " " TO phasevar
taskvar=SPACE(60)
STORE 1 TO pagenova
* Will need to track each unit as done
STORE SPACE(60) TO unitvar
**********************************
STORE 0 TO ordervar
STORE 0 TO countvar
STORE SPACE(60) TO taskvar
* dbf's and variables
DO startplr
SELECT 21
SET ORDER TO unitdodi
ordervar=NUMBER
IF printdes="f" .OR. printdes="p"
        ACTIVATE WINDOW process
ENDIF (printdes="f" .OR. printdes="p")
countvar=1
SELECT 18
SET FILTER TO
SET FILTER TO NUMBER=ordervar
GO TOP
SET ORDER TO phasetas
GO TOP
IF EOF()
        ACTIVATE WINDOW small
        WAIT "ERROR - ORDERTAS.DBF EOF  ORDERVAR= "+ordervar
        DEACTIVATE WINDOW small
```

```
ENDIF (EOF())
CLEAR
* clear previous dodic calc entries
SELECT 19
DO WHILE .NOT. EOF()
    REPLACE  taskorg_qt WITH 0
    REPLACE order_qt WITH 0
    REPLACE order_qttl WITH 0
    REPLACE taskorg_cb WITH 0
    REPLACE order_cb WITH 0
    REPLACE order_cbtl WITH 0
    SKIP
ENDDO (.NOT. EOF())
GO TOP
DO WHILE .T.
    SELECT 21
    SET FILTER TO
    SET FILTER TO taskorgnam=ordertas->taskorgnam .AND. phase=ordertas->phase
    GO TOP
    IF EOF()
        ? "Taskorgnam "+ordertas->taskorgnam
        ? "Phase "+STR(ordertas->phase)
        WAIT
        CLOSE DATA
        RETURN
    ENDIF (EOF())
    IF printdes="s" .AND. detailva="NO"
        CLEAR
    ENDIF (printdes="s" .AND. detailva="NO")
    DO reporthd
    IF PROW()>varpagel-9 .AND. printdes="p"
        ? "PAGE "+LTRIM(STR(pagenova))
        pagenova=pagenova+1
        EJECT
    ENDIF (PROW()>varpagel-9 .AND. printdes="p")
    IF printdes="s"                 &&.AND. row()>20
        ?
        WAIT
        CLEAR
    ENDIF (printdes="s"                 &&.AND. row()>20)
    IF printdes<>"s"
        ?
        ?
    ENDIF (printdes<>"s")
    ? taskorgnam
    ?"_____         "
    ?
    taskvar=ammoest->taskorgnam
    STORE RECNO() TO recnovar
    DO WHILE unitqty<=ammoest->qty
        unitvar=ammoest->unitname
        qtycount=ammoest->qty
        IF detailva="YES"
            ?
            ?
            ? src+" "+unitvar
            ? "          "+TRANSFORM(unitqty,"99")+" OF "+TRANSFORM(qtycount,"99")
            ?? "                        LIN by LIN CONSUMPTION"
            IF printdes="p" .OR. printdes="f"
```

266

```
          ? "LIN        QTY WEAPON NOMENCLATURE          DODIC      DODIC
NOMENCLATURE             ROUNDS      CBT LOAD         WEIGHT         CUBE"
          ? "---        --- --------------------          -----
------------------               ------        --------        ------          ----"
          ELSE
          ? "LIN        QTY WEAPON NOMENCLATURE          DODIC      DODIC
NOMENCLATURE"
          ? "---        --- --------------------          -----
------------------ "
          ? SPACE(20)+"              ROUNDS        CBT LOAD        WEIGHT
CUBE"
          ? SPACE(20)+"              ------        --------        ------
----"
        ENDIF (printdes="p" .OR. printdes="f")
      ENDIF (detailva="YES")
      ****
      IF printdes="s" .AND. detailva="NO"
         CLEAR
      ENDIF (printdes="s" .AND. detailva="NO")
      DO WHILE .NOT. EOF()
         DO linamm
         DO ammocalc
         SKIP
         IF unitname<>unitvar
            IF vttlva="uto"
               DO punitamm
            ENDIF (vttlva="uto")
            EXIT
         ENDIF (unitname<>unitvar)
      ENDDO (.NOT. EOF())
      IF unitqty<qtycount
         unitqty=unitqty+1
         GO recnovar
         LOOP
      ELSE
         IF .NOT. EOF()
            STORE RECNO() TO recnovar
            unitvar=ammoest->unitname
            unitqty=1
            qtycount=ammoest->qty
         ELSE
            EXIT
         ENDIF (.NOT. EOF())
      ENDIF (unitqty<qtycount)
   ENDDO (unitqty<=ammoest->qty)
   unitqty=1
   IF vttlva<>"o"
      DO ptaskamm
   ENDIF (vttlva<>"o")
   SELECT 18
   SKIP
   IF LTRIM(STR(phase))<>phasevar
      DO pordeamm
   ENDIF (LTRIM(STR(phase))<>phasevar)
   STORE LTRIM(STR(phase))TO phasevar
   SELECT 18
   IF EOF()
      EXIT
   ENDIF (EOF())
ENDDO                    && ordertas (.T.)
```

267

```
* List Order totals
IF printdes<>"s"
    ?
    ?
ENDIF (printdes<>"s")
? "        ORDER AMMO CONSUMPTION TOTAL"
IF printdes<>"s"
    ? "                                            DODIC    DODIC NOMENCLATURE
      ROUNDS                          WEIGHT      CUBE"
    ? "                                            -----    ------------------
      ------                          ------      ----"
    ?
ELSE
    ? "DODIC      DODIC NOMENCLATURE"
    ? "-----      ------------------"
    ? SPACE(20)+"                ROUNDS                    WEIGHT        CUBE"
    ? SPACE(20)+"                ------                    ------        ----"
ENDIF (printdes<>"s")
SELECT 19
GO TOP
DO WHILE .NOT. EOF()
    IF dodic->order_qttl>0
        ?
        IF printdes<>"s"
            ?? SPACE(38)
        ENDIF (printdes<>"s")
        ??dodic->dodic+" "+SUBSTR(dodic->dodic_nom,1,20)
        IF printdes="s"
            ? SPACE(20)
        ENDIF (printdes="s")
        ?? " "+TRANSFORM(dodic->order_qttl,"@R 99,999,999,999")
        ?? "                    "
        ?? " "+TRANSFORM(dodic->order_qttl*dodic->weight,"99,999,999.99")
        ?? TRANSFORM(dodic->order_qttl*dodic->cube,"9,999,999.99")
        attlwgt=attlwgt+dodic->order_qttl*dodic->weight
        attlcube=attlcube+dodic->order_qttl*dodic->cube
        REPLACE dodic->order_qttl WITH 0
    ENDIF (dodic->order_qttl>0)
    IF printdes="s" .AND. ROW()>20
        ?
        WAIT
        CLEAR
    ENDIF (printdes="s" .AND. ROW()>20)
    IF PROW()>varpagel-5 .AND. printdes="p"
        ? "PAGE "+LTRIM(STR(pagenova))
        pagenova=pagenova+1
        EJECT
    ENDIF (PROW()>varpagel-5 .AND. printdes="p")
    SKIP
ENDDO (.NOT. EOF())
GO TOP
SELECT 21
IF PROW()>varpagel-5 .AND. printdes="p"
    ? "PAGE "+LTRIM(STR(pagenova))
    pagenova=pagenova+1
    EJECT
ENDIF (PROW()>varpagel-5 .AND. printdes="p")
?
? SPACE(80)+"Totals: "
? SPACE(80)+"        Weight: "+TRANSFORM(attlwgt,"999,999,999.99")
```

268

```
? SPACE(80)+"    Short Tons: "+TRANSFORM(attlwgt/2000,"999,999,999.99")
? SPACE(80)+"         Cube: "+TRANSFORM(attlcube,"999,999,999.99")
* Print last page number
IF PROW()>1 .AND. printdes="p"
   DO WHILE PROW()<varpagel-1
      ?
   ENDDO (PROW()<varpagel-1)
   ? "PAGE "+LTRIM(STR(pagenova))
ENDIF (PROW()>1 .AND. printdes="p")
SET PRINT OFF
SET DEVICE TO SCREEN
IF printdes="s"
   ?
   WAIT
   CLEAR
ENDIF (printdes="s")
IF printdes="f"
   ?
   ? "END OF REPORT"
   SET ALTERNATE OFF
   SET DEVICE TO SCREEN
   DEACTIVATE WINDOW process
ENDIF (printdes="f")
IF printdes="p"
   DEACTIVATE WINDOW process
ENDIF (printdes="p")
SELECT 18
SET FILTER TO
SELECT 21
SET FILTER TO
*!********************************************************************
*!
*!       Procedure: STARTPLR
*!
*!       Called by: PRINTPOL.PRG
*!                : PRINTAMM.PRG
*!                : PRINTOTH.PRG
*!                : PRIN_PRO              (procedure in SYSREP.PRG)
*!
*!           Calls: STARTHLP.PRG
*!
*!            Uses: TASKORG.DBF
*!                : CSRCLIST.DBF
*!                : ORDERLST.DBF
*!                : CPROFILE.DBF
*!                : PROFILES.DBF
*!                : POLRATE.DBF
*!                : POSTURE.DBF
*!                : INTENSTY.DBF
*!                : POLWGT.DBF
*!                : CSRCLINQ.DBF
*!                : ORDERTAS.DBF
*!                : DODIC.DBF
*!                : POLEST.DBF
*!                : AMMOEST.DBF
*!                : AMMORATE.DBF
*!                : MEDICAL.DBF
*!                : WATER.DBF
*!                : CLASS_I.DBF
*!                : CL_OTHER.DBF
```

269

```
*!
*!       MDX files: TASKORG.MDX
*!                : CSRCLIST.MDX
*!                : ORDERLST.MDX
*!                : CPROFILE.MDX
*!                : PROFILES.MDX
*!                : POLRATE.MDX
*!                : INTENSTY.MDX
*!                : POLWGT.MDX
*!                : CSRCLINQ.MDX
*!                : ORDERTAS.MDX
*!                : DODIC.MDX
*!                : POLEST.MDX
*!                : AMMOEST.MDX
*!                : AMMORATE.MDX
*!                : MEDICAL.MDX
*!                : WATER.MDX
*!                : CLASS_I.MDX
*!                : CL_OTHER.MDX
*!
*!***************************************************************************
PROCEDURE startplr
* OPEN FILES
CLOSE DATA
STORE 1 TO statusvar
DO starthlp
SELECT 1
USE taskorg
SELECT 2
USE csrclist
SELECT 3
USE orderlst
SELECT 4
USE cprofile ORDER profilenam
SELECT 7
USE profiles
SELECT 8
USE ammorate ORDER lin
SELECT 17
USE posture
SELECT 10
USE intensty
SELECT 12
USE polwgt ORDER pol_type
SELECT 14
USE csrclinq
SELECT 18
USE ordertas
SELECT 19
USE dodic ORDER dodic
SELECT 21
USE ammoest
* SET RELATIONS
SELECT 18
SET RELATION TO profilenam INTO cprofile
SELECT 3
USE orderlst
SET ORDER TO NUMBER
SEEK ammoest->number
SELECT 14
```

270

```
SET ORDER TO lin
SELECT 21
SET ORDER TO lin
SET RELATION TO lin INTO csrclinq
SET ORDER TO dodic
SET RELATION TO dodic INTO dodic
SELECT 21
SET ORDER TO unitdodi
*!***********************************************************************
*!
*!        Procedure: AMMOCALC
*!
*!        Called by: CALCAMMO.PRG
*!                 : PRIN_PRO              (procedure in SYSREP.PRG)
*!
*!***********************************************************************
PROCEDURE ammocalc
REPLACE dodic->order_qt WITH dodic->order_qt+ammoest->calc_qty
REPLACE dodic->order_qttl WITH dodic->order_qttl+ammoest->calc_qty
REPLACE dodic->order_cb WITH dodic->order_cb+ammoest->cbt_loadqt
REPLACE dodic->order_cbtl WITH dodic->order_cbtl+ammoest->cbt_loadqt
REPLACE dodic->taskorg_qt WITH dodic->taskorg_qt+ammoest->calc_qty
REPLACE dodic->taskorg_cb WITH dodic->taskorg_cb+ammoest->cbt_loadqt
REPLACE dodic->unit_qt WITH dodic->unit_qt+ammoest->calc_qty
REPLACE dodic->unit_cb WITH dodic->unit_cb+cbt_loadqt
*!***********************************************************************
*!
*!        Procedure: PUNITAMM
*!
*!        Called by: PRIN_PRO              (procedure in SYSREP.PRG)
*!
*!***********************************************************************
PROCEDURE punitamm
IF printdes<>"s"
    ?
    ?
ENDIF (printdes<>"s")
IF PROW()>varpagel-10 .AND. printdes="p"
    ? "PAGE "+LTRIM(STR(pagenova))
    pagenova=pagenova+1
    EJECT
ENDIF (PROW()>varpagel-10 .AND. printdes="p")
IF printdes="s" .AND. detailva="NO"
    CLEAR
ENDIF (printdes="s" .AND. detailva="NO")
?
? src+" "+unitvar
? "         "+TRANSFORM(unitqty,"99")+" OF "+TRANSFORM(qtycount,"99")
?? "                  DODIC SUMMARY SECTION"
IF printdes<>"s"
    ? "                                        DODIC     DODIC NOMENCLATURE
    ROUNDS       CBT LOAD      WEIGHT       CUBE"
    ? "                                        -----     --------------------
    ------       --------      ------       ----"
    ?
ELSE
    ? "DODIC      DODIC NOMENCLATURE"
    ? "-----      --------------------"
    ? SPACE(20)+"          ROUNDS     CBT LOAD      WEIGHT          CUBE"
    ? SPACE(20)+"          ------     --------      ------          ----"
```

271

```
ENDIF (printdes<>"s")
SELECT 19
GO TOP
DO WHILE .NOT. EOF()
    IF dodic->unit_qt>0
        ?

        IF printdes<>"s"
            ?? SPACE(38)
        ENDIF (printdes<>"s")
        ?? dodic->dodic+" "+SUBSTR(dodic->dodic_nom,1,20)
        IF printdes="s"
            ? SPACE(20)
        ENDIF (printdes="s")
        ?? " "+TRANSFORM(dodic->unit_qt,"@R 99,999,999,999")
        ?? " "+TRANSFORM(dodic->unit_cb,"@R 9,999,999,999")
        ?? " "+TRANSFORM(dodic->unit_qt*dodic->weight,"9,999,999.99")
        ?? "  "+TRANSFORM(dodic->unit_qt*dodic->cube,"999,999.99")
        * Keep track of unit ammo weight and cube
        ottlwgt=ottlwgt+dodic->order_qttl*dodic->weight
        ottlcube=ottlcube+dodic->order_qttl*dodic->cube
        REPLACE dodic->unit_qt WITH 0
        REPLACE dodic->unit_cb WITH 0
    ENDIF (dodic->unit_qt>0)
    IF printdes="s" .AND. ROW()>20
        ?
        WAIT
        CLEAR
    ENDIF (printdes="s" .AND. ROW()>20)
    IF PROW()>varpagel-5 .AND. printdes="p"
        ? "PAGE "+LTRIM(STR(pagenova))
        pagenova=pagenova+1
        EJECT
    ENDIF (PROW()>varpagel-5 .AND. printdes="p")
    SKIP
ENDDO (.NOT. EOF())
* print Unit weight and cube totals
? SPACE(80)+"Unit Totals: "
? SPACE(80)+"        Weight: "+TRANSFORM(ottlwgt,"999,999,999.99")
? SPACE(80)+"    Short Tons: "+TRANSFORM(ottlwgt/2000,"999,999,999.99")
? SPACE(80)+"          Cube: "+TRANSFORM(ottlcube,"999,999,999.99")
ottlwgt=0
ottlcube=0
GO TOP
SELECT 21
IF printdes="s" .AND. ROW()>20
    ?
    WAIT
    CLEAR
ENDIF (printdes="s" .AND. ROW()>20)
IF PROW()>varpagel-5 .AND. printdes="p"
    ? "PAGE "+LTRIM(STR(pagenova))
    pagenova=pagenova+1
    EJECT
ENDIF (PROW()>varpagel-5 .AND. printdes="p")
*!*****************************************************************
*!
*!      Procedure: PTASKAMM
*!
*!      Called by: PRIN_PRO              (procedure in SYSREP.PRG)
*!
```

```
*!***********************************************************************
PROCEDURE ptaskamm
?
?
IF PROW()>varpagel-9 .AND. printdes="p"
    ? "PAGE "+LTRIM(STR(pagenova))
    pagenova=pagenova+1
    EJECT
ENDIF (PROW()>varpagel-9 .AND. printdes="p")
IF printdes="s".AND. ROW()>10 .AND. detailva="YES"
    WAIT
    CLEAR
ENDIF (printdes="s".AND. ROW()>10 .AND. detailva="YES")
****
IF printdes="s" .AND. detailva="NO"
    CLEAR
ENDIF (printdes="s" .AND. detailva="NO")
* List Task Org totals
? taskvar
?
? "TASK ORG AMMO FOR PHASE "+phasevar
IF printdes<>"s"
    ? "                                            DODIC    DODIC NOMENCLATURE
    ROUNDS       CBT LOAD       WEIGHT       CUBE"
    ? "                                            -----    --------------------
    ------       --------       ------       ----"
    ?
ELSE
    ? "DODIC    DODIC NOMENCLATURE"
    ? "-----    --------------------"
    ? SPACE(20)+"           ROUNDS       CBT LOAD       WEIGHT       CUBE"
    ? SPACE(20)+"           ------       --------       ------       ----"
ENDIF (printdes<>"s")
SELECT 19
GO TOP
DO WHILE .NOT. EOF()
    IF dodic->taskorg_qt>0
        ?
        IF printdes<>"s"
            ?? SPACE(38)
        ENDIF (printdes<>"s")
        ?? dodic->dodic+" "+SUBSTR(dodic->dodic_nom,1,20)
        IF printdes="s"
            ? SPACE(20)
        ENDIF (printdes="s")
        ?? " "+TRANSFORM(dodic->taskorg_qt,"@R 99,999,999,999")
        ?? " "+TRANSFORM(dodic->taskorg_cb,"@R 9,999,999,999")
        ?? " "+TRANSFORM(dodic->taskorg_qt*dodic->weight,"9,999,999.99")
        ?? "   "+TRANSFORM(dodic->taskorg_qt*dodic->cube,"999,999.99")
        *?
        * Keep track of Task Org ammo weight and cube
        tttlwgt=tttlwgt+dodic->taskorg_qt*dodic->weight
        tttlcube=tttlcube+dodic->taskorg_qt*dodic->cube
        REPLACE dodic->taskorg_qt WITH 0
        REPLACE dodic->taskorg_cb WITH 0
    ENDIF (dodic->taskorg_qt>0)
    IF printdes="s" .AND. ROW()>20
        ?
        WAIT
        CLEAR
```

273

```
        ENDIF (printdes="s" .AND. ROW()>20)
        IF PROW()>varpagel-5 .AND. printdes="p"
            ? "PAGE "+LTRIM(STR(pagenova))
            pagenova=pagenova+1
            EJECT
        ENDIF (PROW()>varpagel-5 .AND. printdes="p")
        SKIP
ENDDO (.NOT. EOF())
?
? SPACE(80)+"Task Force Totals This Phase: "
? SPACE(80)+"         Weight: "+TRANSFORM(tttlwgt,"999,999,999.99")
? SPACE(80)+"     Short Tons: "+TRANSFORM(tttlwgt/2000,"999,999,999.99")
? SPACE(80)+"           Cube: "+TRANSFORM(tttlcube,"999,999,999.99")
* reinitialize task force weight and cube variables
tttlwgt=0
tttlcube=0
GO TOP
SELECT 21
?
IF printdes="s"
    ?
    WAIT
    CLEAR
ENDIF (printdes="s")
*!********************************************************************
*!
*!      Procedure: PORDEAMM
*!
*!      Called by: PRIN_PRO                (procedure in SYSREP.PRG)
*!
*!********************************************************************
PROCEDURE pordeamm
?
?
IF PROW()>varpagel-9 .AND. printdes="p"
    ? "PAGE "+LTRIM(STR(pagenova))
    pagenova=pagenova+1
    EJECT
ENDIF (PROW()>varpagel-9 .AND. printdes="p")
* List Order totals this phase
CLEAR
? "        ORDER AMMO PHASE "+phasevar
IF printdes<>"s"
    ? "                                                DODIC    DODIC NOMENCLATURE
    ROUNDS         CBT LOAD        WEIGHT        CUBE"
    ? "                                                -----    ------------------
    ------         --------        ------        ----"
    ?
ELSE
    ? "DODIC      DODIC NOMENCLATURE"
    ? "-----      ------------------"
    ? SPACE(20)+"            ROUNDS        CBT LOAD        WEIGHT          CUBE"
    ? SPACE(20)+"            ------        --------        ------          ----"
ENDIF (printdes<>"s")
SELECT 19
GO TOP
DO WHILE .NOT. EOF()
    IF dodic->order_qt>0
        ?
        IF printdes<>"s"
```

274

```
            ?? SPACE(38)
        ENDIF (printdes<>"s")
        ?? dodic->dodic+" "+SUBSTR(dodic->dodic_nom,1,20)
        IF printdes="s"
            ? SPACE(20)
        ENDIF (printdes="s")
        ?? " "+TRANSFORM(dodic->order_qt,"@R 99,999,999,999")
        ?? " "+TRANSFORM(dodic->order_cb,"@R 9,999,999,999")
        ?? " "+TRANSFORM(dodic->order_qt*dodic->weight,"9,999,999.99")
        ?? "  "+TRANSFORM(dodic->order_qt*dodic->cube,"999,999.99")
        * Keep track of Order ammo weight & cube this Phase
        pttlwgt=pttlwgt+dodic->order_qt*dodic->weight
        pttlcube=pttlcube+dodic->order_qt*dodic->cube
        REPLACE dodic->order_qt WITH 0
        REPLACE dodic->order_cb WITH 0
    ENDIF (dodic->order_qt>0)
    SKIP
    IF printdes="s" .AND. ROW()>17
        ?
        WAIT
        CLEAR
    ENDIF (printdes="s" .AND. ROW()>17)
    IF PROW()>varpagel-5 .AND. printdes="p"
        ? "PAGE "+LTRIM(STR(pagenova))
        pagenova=pagenova+1
        EJECT
    ENDIF (PROW()>varpagel-5 .AND. printdes="p")
ENDDO (.NOT. EOF())
?
? SPACE(80)+"Order Totals This Phase: "
? SPACE(80)+"          Weight: "+TRANSFORM(pttlwgt,"999,999,999.99")
? SPACE(80)+"      Short Tons: "+TRANSFORM(pttlwgt/2000,"999,999,999.99")
? SPACE(80)+"            Cube: "+TRANSFORM(pttlcube,"999,999,999.99")
* reinitialize task force weight and cube variables
pttlwgt=0
pttlcube=0
GO TOP
SELECT 21
?
IF printdes="s"
    ?
    WAIT
    CLEAR
ENDIF (printdes="s")
*!*********************************************************************
*!
*!      Procedure: LINAMM
*!
*!      Called by: PRIN_PRO          (procedure in SYSREP.PRG)
*!
*!*********************************************************************
PROCEDURE linamm
* Do one LIN at a time
IF detailva="YES"
    IF lin_nomen<>nomenvar
        ? lin+" "+TRANSFORM(quantity,"9999")+" "+SUBSTR(lin_nomen,1,25)
    ELSE
        ? SPACE(6)+" "+SPACE(4)+" "+SPACE(25)
    ENDIF (lin_nomen<>nomenvar)
    ?? " "+dodic+" "+SUBSTR(dodic->dodic_nom,1,20)
```

275

```
      IF printdes="s"
         ? SPACE(20)
      ENDIF (printdes="s")
      ?? " "+TRANSFORM(calc_qty,"@R 99,999,999,999")
      ?? " "+TRANSFORM(cbt_loadqt,"@R 9,999,999,999")
      ?? " "+TRANSFORM(calc_qty*dodic->weight,"9,999,999.99")
      ?? "  "+TRANSFORM(calc_qty*dodic->cube,"999,999.99")
      nomenvar=lin_nomen
      *         ? "                        USE: "+amm1var+"  CBT LOAD: "+amm2var
   ENDIF (detailva="YES")
   IF PROW()>varpagel-5 .AND. printdes="p"
      ? "PAGE "+LTRIM(STR(pagenova))
      pagenova=pagenova+1
      EJECT
   ENDIF (PROW()>varpagel-5 .AND. printdes="p")
   IF printdes="s" .AND. ROW()>20
      WAIT
      CLEAR
   ENDIF (printdes="s" .AND. ROW()>20)
*!******************************************************************
*!
*!      Procedure: REPORTHD
*!
*!      Called by: PRIN_PRO             (procedure in SYSREP.PRG)
*!
*!******************************************************************
PROCEDURE reporthd
IF printdes<>"s"
   ?
   ?
ENDIF (printdes<>"s")
IF PROW()>varpagel-4 .AND. printdes="p"
   ? "PAGE "+LTRIM(STR(pagenova))
   pagenova=pagenova+1
   EJECT
ENDIF (PROW()>varpagel-4 .AND. printdes="p")
IF printdes="s".AND. ROW()>18
   WAIT
   CLEAR
ENDIF (printdes="s".AND. ROW()>18)
* Need to choose phase length
plenvar="PHASE"+LTRIM(STR(ammoest->phase))+"LEN"
? ammoest->number+" "+orderlst->name
IF printdes="s"
   ?
ENDIF (printdes="s")
?? "  ESTIMATED AMMUNITION CONSUMPTION REPORT as of "+DTOC(DATE())
? ammoest->taskorgnam
phasevar=LTRIM(STR(ammoest->phase))
? "Phase: "+phasevar+"  Strength "+LTRIM(STR(ordertas->strength))+"%"
?? "  Phase Length: "
DO CASE
CASE plenvar="PHASE1LEN"
   ?? LTRIM(STR(orderlst->phase1len))
CASE plenvar="PHASE2LEN"
   ?? LTRIM(STR(orderlst->phase2len))
CASE plenvar="PHASE3LEN"
   ?? LTRIM(STR(orderlst->phase3len))
CASE plenvar="PHASE4LEN"
   ?? LTRIM(STR(orderlst->phase4len))
```

```
CASE plenvar="PHASE5LEN"
    ?? LTRIM(STR(orderlst->phase5len))
CASE plenvar="PHASE6LEN"
    ?? LTRIM(STR(orderlst->phase6len))
CASE plenvar="PHASE7LEN"
    ?? LTRIM(STR(orderlst->phase7len))
CASE plenvar="PHASE8LEN"
    ?? LTRIM(STR(orderlst->phase8len))
CASE plenvar="PHASE9LEN"
    ?? LTRIM(STR(orderlst->phase9len))
ENDCASE
? "Parameter Set: "+ordertas->profilenam+"  "+cprofile->profiledes
****************************************************
? "Posture: "+cprofile->posture
? "Intensity: "+cprofile->intensity+" Factor:
"+TRANSFORM(cprofile->factor,"999.99")
*!*******************************************************************
*!
*!       Procedure: DETA_PRO
*!
*!       Called by: PRINTPOL.PRG
*!                : PRINTAMM.PRG
*!                : PRINTOTH.PRG
*!
*!*******************************************************************
PROCEDURE deta_pro
DO CASE
CASE BAR()=1
    detailva="YES"
    DEACTIVATE POPUP
CASE BAR()=2
    detailva="NO"
    vttlva="uto"
    DEACTIVATE POPUP
CASE BAR()=3
    detailva="NO"
    vttlva="to"
    DEACTIVATE POPUP
CASE BAR()=4
    detailva="NO"
    vttlva="o"
    DEACTIVATE POPUP
ENDCASE
CLEAR
* BAR 1 OF deta_pop PROMPT "LIN by LIN Detail, plus all Totals"
*!*******************************************************************
*!
*!       Procedure: DETA_BAR1
*!
*!*******************************************************************
PROCEDURE deta_bar1
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
   Consumption data is printed for each equipment LIN, including DODICs,
   rounds, basic load, weight of rounds consumed, and cube of rounds
   consumed. Unit, Task Organization, and Order summaries follow LIN by
   LIN  detail.  A seperate listing is printed for each Unit if there are
   more than one of a particular Unit.  This can be a very long report,
   and it may be preferred to print it to a file for use with a word
```

277

```
      processor.
ENDTEXT

RETURN
* BAR 2 OF deta_pop PROMPT "Summary Report - Unit, Task Org, and Order Totals"
*!**************************************************************************
*!
*!      Procedure: DETA_BAR2
*!
*!**************************************************************************
PROCEDURE deta_bar2
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
   Consumption data is printed for each DODIC, including rounds, weight
   of rounds consumed, and cube of rounds consumed by Unit, Task
   Organization, and Order.
ENDTEXT

RETURN
* BAR 3 OF deta_pop PROMPT "Summary Report - Task Org and Order Totals"
*!**************************************************************************
*!
*!      Procedure: DETA_BAR3
*!
*!**************************************************************************
PROCEDURE deta_bar3
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
   Consumption data is printed for each DODIC, including rounds, weight
   of rounds consumed, and cube of rounds consumed by Task Organization
   and Order.
ENDTEXT

RETURN
* BAR 4 OF deta_pop PROMPT "Summary Report - Order Totals"
*!**************************************************************************
*!
*!      Procedure: DETA_BAR4
*!
*!**************************************************************************
PROCEDURE deta_bar4
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
   Consumption data is printed for each DODIC, including rounds, weight
   of rounds consumed, and cube of rounds consumed by the Order.
ENDTEXT

RETURN
* BAR 1 OF prin_pop PROMPT "Print Estimate to Printer"
*!**************************************************************************
*!
*!      Procedure: PRIN_BAR1
*!
*!**************************************************************************
PROCEDURE prin_bar1
@ 9,1 CLEAR TO 16,79
@ 9,1
```

278

```
TEXT
    Make sure printer is on and on-line.
ENDTEXT


RETURN
* BAR 2 OF prin_pop PROMPT "Print Estimate to Screen"
*!**********************************************************************
*!
*!      Procedure: PRIN_BAR2
*!
*!**********************************************************************
PROCEDURE prin_bar2
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
    Prints Reports to the Screen. These estimates can take a long time
    to view this way.  It may be preferred to print to a file and then
    view the file with a work processor.
ENDTEXT


RETURN
* BAR 3 OF prin_pop PROMPT "Print Estimate to ASCII File"
*!**********************************************************************
*!
*!      Procedure: PRIN_BAR3
*!
*!**********************************************************************
PROCEDURE prin_bar3
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
    ASCII files can be viewed and edited with most word processors, or
    with EDIT if you are using a version of DOS 5.0 or higher.
ENDTEXT


RETURN
* BAR 4 OF prin_pop PROMPT "Change Printer Defaults"
*!**********************************************************************
*!
*!      Procedure: PRIN_BAR4
*!
*!**********************************************************************
PROCEDURE prin_bar4
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
    Change printer page length and left margin.
ENDTEXT


RETURN
* BAR 5 OF prin_pop PROMPT "Return to Previous Menu"
*!**********************************************************************
*!
*!      Procedure: PRIN_BAR5
*!
*!**********************************************************************
PROCEDURE prin_bar5
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
```

```
ENDTEXT

RETURN

*: EOF: PRINTAMM.PRG
```

```
*:****************************************************************
*:
*: Procedure file: D:\SURE\PRINTOTH.PRG
*:
*:   Procs & Fncts: OTHECALC
*:                : PUNITOTH
*:                : PTASKOTH
*:                : PORDEOTH
*:
*:          Set by: ORDR_PRO              (procedure in ORDEREP1.PRG)
*:
*:           Calls: DETA_PRO              (procedure in PRINTPOL.PRG)
*:                : STARTPLR              (procedure in PRINTPOL.PRG)
*:                : PRIN_PRO              (procedure in SYSREP.PRG)
*:
*:        Documented 04/16/94 at 11:02                SNAP!  version 5.00
*:****************************************************************
STORE PROGRAM() TO progvar
STORE 0 TO strenvar
STORE "   " TO vttlva
STORE SPACE(60) TO filevar
STORE SPACE(9) TO srcvar
STORE " NO" TO detailva
STORE " " TO printdes
STORE SPACE(9) TO plenvar
STORE 1 TO qtycount
STORE 1 TO unitqty
* water variables
STORE 0 TO u_h2ovar
STORE 0 TO t_h2ovar
STORE 0 TO o_h2ovar
STORE 0 TO oth2ovar
* class iv barrier variables
STORE 0 TO u_ivbvar
STORE 0 TO t_ivbvar
STORE 0 TO o_ivbvar
STORE 0 TO otivbvar
* class iv construction variables
STORE 0 TO u_ivcvar
STORE 0 TO t_ivcvar
STORE 0 TO o_ivcvar
STORE 0 TO otivcvar
* class viii variables
STORE 0 TO u_cl8var
STORE 0 TO t_cl8var
STORE 0 TO o_cl8var
STORE 0 TO otcl8var
CLEAR
DEFINE POPUP deta_pop FROM 1,1
DEFINE BAR 1 OF deta_pop PROMPT "Unit, Task Org, and Order Totals"
DEFINE BAR 2 OF deta_pop PROMPT "Task Org and Order Totals Only"
DEFINE BAR 3 OF deta_pop PROMPT "Order Totals Only"
ON SELECTION POPUP deta_pop DO deta_pro
ON BAR 1 OF deta_pop DO deta_bar1
ON BAR 2 OF deta_pop DO deta_bar2
ON BAR 3 OF deta_pop DO deta_bar3
DO startplr
DEFINE POPUP prin_pop FROM 1,1
DEFINE BAR 1 OF prin_pop PROMPT "Print Estimate to Printer"
DEFINE BAR 2 OF prin_pop PROMPT "Print Estimate to Screen"
```

```
DEFINE BAR 3 OF prin_pop PROMPT "Print Estimate to ASCII File"
DEFINE BAR 4 OF prin_pop PROMPT "Change Printer Defaults"
DEFINE BAR 5 OF prin_pop PROMPT "DONE!"
ON SELECTION POPUP prin_pop DO prin_pro
ON BAR 1 OF prin_pop DO prin_bar1
ON BAR 2 OF prin_pop DO prin_bar2
ON BAR 3 OF prin_pop DO prin_bar3
ON BAR 4 OF prin_pop DO prin_bar4
ON BAR 5 OF prin_pop DO prin_bar5
DO WHILE PROMPT()<>"DONE!"
    ACTIVATE POPUP prin_pop
ENDDO (PROMPT()<>"DONE!")
*!************************************************************************
*!
*!        Procedure: PRIN_PRO
*!
*!        Called by: SYSREP.PRG
*!                 : PRINTPOL.PRG
*!                 : PRINTAMM.PRG
*!                 : PRINTOTH.PRG
*!
*!            Calls: PRNSETUP.PRG
*!                 : STARTPLR          (procedure in PRINTPOL.PRG)
*!                 : REPORTHD          (procedure in PRINTPOL.PRG)
*!                 : TFHEAD            (procedure in PRINTPOL.PRG)
*!                 : LINPOL            (procedure in PRINTPOL.PRG)
*!                 : POLCALC           (procedure in PRINTPOL.PRG)
*!                 : PUNITPOL          (procedure in PRINTPOL.PRG)
*!                 : PTASKPOL          (procedure in PRINTPOL.PRG)
*!                 : PORDEPOL          (procedure in PRINTPOL.PRG)
*!                 : LINAMM            (procedure in PRINTAMM.PRG)
*!                 : AMMOCALC          (procedure in PRINTAMM.PRG)
*!                 : PUNITAMM          (procedure in PRINTAMM.PRG)
*!                 : PTASKAMM          (procedure in PRINTAMM.PRG)
*!                 : PORDEAMM          (procedure in PRINTAMM.PRG)
*!                 : OTHECALC          (procedure in PRINTOTH.PRG)
*!                 : PUNITOTH          (procedure in PRINTOTH.PRG)
*!                 : PTASKOTH          (procedure in PRINTOTH.PRG)
*!                 : PORDEOTH          (procedure in PRINTOTH.PRG)
*!
*!************************************************************************
PROCEDURE prin_pro
DO CASE
CASE BAR()=1
    printdes="p"
    DO CASE
    CASE vprintva="Dot Matrix"
        ??? CHR(27)+CHR(15)
    CASE vprintva="Laser"
    CASE vprintva="DeskJet"
        ??? CHR(27)+"(s16.66H"
    ENDCASE
CASE BAR()=2
    printdes="s"
    CLEAR
CASE BAR()=3
    printdes="f"
    ACTIVATE WINDOW process
    @ 1,1 SAY "Enter path and file name."
    @ 3,1 GET filevar
```

282

```
        READ
        *filevar=LTRIM(TRIM(filevar))
        SET ALTERNATE TO &filevar
        DEACTIVATE WINDOW process
     CASE BAR()=4
        DO prnsetup
     CASE BAR()=5
        DEACTIVATE POPUP
        RETURN
     ENDCASE
     CLEAR
     ACTIVATE POPUP deta_pop
     * Set according to destination
     DO CASE
     CASE printdes="p"
        SET DEVICE TO PRINT
        SET PRINT ON
     CASE printdes="s"
     CASE printdes="f"
        SET ALTERNATE ON
     ENDCASE
     * Initialize Variables
     STORE " " TO phasevar
     taskvar=SPACE(60)
     STORE 1 TO pagenova
     * Will need to track each unit as done
     STORE SPACE(60) TO unitvar
     STORE 0 TO ordervar
     STORE 0 TO countvar
     STORE SPACE(60) TO taskvar
     * dbf's and variables
     DO startplr
     SELECT 26
     ordervar=NUMBER
     IF printdes="f" .OR. printdes="p"
        ACTIVATE WINDOW process
     ENDIF (printdes="f" .OR. printdes="p")
     countvar=1
     SELECT 18
     SET FILTER TO
     SET FILTER TO NUMBER=ordervar
     SET ORDER TO phasetas
     GO TOP
     IF EOF()
        ACTIVATE WINDOW small
        WAIT "ERROR EOF PRINTOTH.PRG"
        DEACTIVATE WINDOW small
     ENDIF (EOF())
     CLEAR
     * clear previous class_i calc entries
     SELECT 24
     DO WHILE .NOT. EOF()
        REPLACE  unit_qty WITH 0
        REPLACE task_org_q WITH 0
        REPLACE order_qty WITH 0
        REPLACE order_qty WITH 0
        REPLACE order_ttl WITH 0
        SKIP
     ENDDO (.NOT. EOF())
     GO TOP
```

283

```
DO WHILE .T.                      && ordertas
    SELECT 26
    SET FILTER TO
    SET FILTER TO taskorgnam=ordertas->taskorgnam .AND. phase=ordertas->phase
    GO TOP
    IF printdes="s" .AND. detailva="NO"
       CLEAR
    ENDIF (printdes="s" .AND. detailva="NO")
    DO reporthd
    IF PROW()>varpagel-9 .AND. printdes="p"
       ? "PAGE "+LTRIM(STR(pagenova))
       pagenova=pagenova+1
       EJECT
    ENDIF (PROW()>varpagel-9 .AND. printdes="p")
    IF printdes="s"                    &&.AND. row()>20
       ?
       WAIT
       CLEAR
    ENDIF (printdes="s"                &&.AND. row()>20)
    taskvar=cl_other->taskorgnam
    unitvar=cl_other->unitname
    strenvar=cl_other->strength
    srcvar=cl_other->src
    IF printdes="s" .AND. detailva="NO"
       CLEAR
    ENDIF (printdes="s" .AND. detailva="NO")
    STORE RECNO() TO recnovar
    unitqty=1
    unitvar=cl_other->unitname
    strenvar=cl_other->strength
    srcvar=cl_other->src
    qtycount=cl_other->qty
    DO WHILE .NOT. EOF()
       DO othecalc
       SELECT 26
       IF vttlva="uto"
          DO punitoth
       ENDIF (vttlva="uto")
       SELECT 26
       IF unitqty<qtycount
          unitqty=unitqty+1
          LOOP
       ENDIF (unitqty<qtycount)
       SKIP
       IF EOF()
          EXIT
       ELSE
          unitqty=1
          qtycount=cl_other->qty
          unitvar=cl_other->unitname
          strenvar=cl_other->strength
          srcvar=cl_other->src
          STORE RECNO() TO recnovar
       ENDIF (EOF())
    ENDDO (.NOT. EOF())
    IF vttlva<>"o"
       DO ptaskoth
    ENDIF (vttlva<>"o")
    SELECT 18
    SKIP
```

284

```
        IF LTRIM(STR(phase))<>phasevar
            DO pordeoth
        ENDIF (LTRIM(STR(phase))<>phasevar)
        STORE LTRIM(STR(phase))TO phasevar
        SELECT 18
        IF EOF()
            EXIT
        ENDIF (EOF())
ENDDO                     && ordertas (.T.              && ordertas) (.T.
        && ordertas)
* List Order totals
IF printdes<>"s"
    ?
    ?
ENDIF (printdes<>"s")
? "       ORDER CONSUMPTION TOTALS"
?
?
SELECT 24
GO TOP
DO WHILE .NOT. EOF()
    ?? TRANSFORM(class_i->order_ttl,"@R 999,999,999")+" "
    REPLACE class_i->order_ttl WITH 0
    SKIP
ENDDO (.NOT. EOF())
GO TOP
IF printdes="s"
    ?
ENDIF (printdes="s")
* Water
?? TRANSFORM(oth2ovar, "@R 999,999,999")+" "
oth2ovar=0
* CLASS IV
?? TRANSFORM(otivbvar, "@R 999,999,999")+" "
?? TRANSFORM(otivcvar, "@R 999,999,999")+" "
otivbvar=0
otivcvar=0
* Class 8
?? TRANSFORM(otcl8var, "@R 999,999,999")
otcl8var=0
* Print last page number
IF PROW()>1 .AND. printdes="p"
    DO WHILE PROW()<varpagel-2
        ?
    ENDDO (PROW()<varpagel-2)
    ? "PAGE "+LTRIM(STR(pagenova))+" END OF REPORT"
ENDIF (PROW()>1 .AND. printdes="p")
SET PRINT OFF
SET DEVICE TO SCREEN
IF printdes="s"
    ?
    WAIT
    CLEAR
ENDIF (printdes="s")
IF printdes="f"
    ?
    ? "END OF REPORT"
    SET ALTERNATE OFF
    SET DEVICE TO SCREEN
    DEACTIVATE WINDOW process
```

```
ENDIF (printdes="f")
IF printdes="p"
    DEACTIVATE WINDOW process
ENDIF (printdes="p")
SELECT 18
SET FILTER TO
*!********************************************************************
*!
*!          Procedure: STARTPLR
*!
*!          Called by: PRINTPOL.PRG
*!                   : PRINTAMM.PRG
*!                   : PRINTOTH.PRG
*!                   : PRIN_PRO               (procedure in SYSREP.PRG)
*!
*!              Calls: STARTHLP.PRG
*!
*!               Uses: TASKORG.DBF
*!                   : CSRCLIST.DBF
*!                   : ORDERLST.DBF
*!                   : CPROFILE.DBF
*!                   : PROFILES.DBF
*!                   : POLRATE.DBF
*!                   : POSTURE.DBF
*!                   : INTENSTY.DBF
*!                   : POLWGT.DBF
*!                   : CSRCLINQ.DBF
*!                   : ORDERTAS.DBF
*!                   : DODIC.DBF
*!                   : POLEST.DBF
*!                   : AMMOEST.DBF
*!                   : AMMORATE.DBF
*!                   : MEDICAL.DBF
*!                   : WATER.DBF
*!                   : CLASS_I.DBF
*!                   : CL_OTHER.DBF
*!
*!          MDX files: TASKORG.MDX
*!                   : CSRCLIST.MDX
*!                   : ORDERLST.MDX
*!                   : CPROFILE.MDX
*!                   : PROFILES.MDX
*!                   : POLRATE.MDX
*!                   : INTENSTY.MDX
*!                   : POLWGT.MDX
*!                   : CSRCLINQ.MDX
*!                   : ORDERTAS.MDX
*!                   : DODIC.MDX
*!                   : POLEST.MDX
*!                   : AMMOEST.MDX
*!                   : AMMORATE.MDX
*!                   : MEDICAL.MDX
*!                   : WATER.MDX
*!                   : CLASS_I.MDX
*!                   : CL_OTHER.MDX
*!
*!********************************************************************
PROCEDURE startplr
* OPEN FILES
CLOSE DATA
```

```
STORE 1 TO statusvar
DO starthlp
SELECT 1
USE taskorg
SELECT 2
USE csrclist
SELECT 3
USE orderlst
SELECT 4
USE cprofile ORDER profilenam
SELECT 7
USE profiles
SELECT 17
USE posture
SELECT 10
USE intensty
SELECT 14
USE csrclinq
SELECT 18
USE ordertas
SELECT 22
USE medical
SELECT 23
USE water
SELECT 24
USE class_i ORDER ration
SELECT 26
USE cl_other
* SET RELATIONS
SELECT 18
SET RELATION TO profilenam INTO cprofile
SELECT 3
USE orderlst
SET ORDER TO NUMBER
SEEK cl_other->number
*!*******************************************************************
*!
*!       Procedure: OTHECALC
*!
*!       Called by: PRIN_PRO            (procedure in SYSREP.PRG)
*!
*!*******************************************************************
PROCEDURE othecalc
* Track Class I
SELECT 24                  && Class_I
* 1st Meal
SEEK cprofile->cl_i_1
REPLACE unit_qty WITH unit_qty+cl_other->cl_i_qty
REPLACE task_org_q WITH task_org_q+cl_other->cl_i_qty
REPLACE order_qty WITH order_qty+cl_other->cl_i_qty
REPLACE order_ttl WITH order_ttl+cl_other->cl_i_qty
* 2nd Meal
SEEK cprofile->cl_i_2
REPLACE unit_qty WITH unit_qty+cl_other->cl_i_qty
REPLACE task_org_q WITH task_org_q+cl_other->cl_i_qty
REPLACE order_qty WITH order_qty+cl_other->cl_i_qty
REPLACE order_ttl WITH order_ttl+cl_other->cl_i_qty
* 3rd Meal
SEEK cprofile->cl_i_3
REPLACE unit_qty WITH unit_qty+cl_other->cl_i_qty
```

287

```
REPLACE task_org_q WITH task_org_q+cl_other->cl_i_qty
REPLACE order_qty WITH order_qty+cl_other->cl_i_qty
REPLACE order_ttl WITH order_ttl+cl_other->cl_i_qty
* Track H2o
u_h2ovar=u_h2ovar+cl_other->h2o_qty
t_h2ovar=t_h2ovar+cl_other->h2o_qty
o_h2ovar=o_h2ovar+cl_other->h2o_qty
oth2ovar=oth2ovar+cl_other->h2o_qty
* Track class iv barrier variables
u_ivbvar=u_ivbvar+cl_other->cl_ivbqty
t_ivbvar=t_ivbvar+cl_other->cl_ivbqty
o_ivbvar=o_ivbvar+cl_other->cl_ivbqty
otivbvar=otivbvar+cl_other->cl_ivbqty
* Track class iv construction variables
u_ivcvar=u_ivcvar+cl_other->cl_ivcqty
t_ivcvar=t_ivcvar+cl_other->cl_ivcqty
o_ivcvar=o_ivcvar+cl_other->cl_ivcqty
otivcvar=otivcvar+cl_other->cl_ivcqty
* Track class viii variables
u_cl8var=u_cl8var+cl_other->cl_viii_qt
t_cl8var=t_cl8var+cl_other->cl_viii_qt
o_cl8var=o_cl8var+cl_other->cl_viii_qt
otcl8var=otcl8var+cl_other->cl_viii_qt
*SELECT 21
RETURN
*!*******************************************************************
*!
*!        Procedure: PUNITOTH
*!
*!        Called by: PRIN_PRO              (procedure in SYSREP.PRG)
*!
*!*******************************************************************
PROCEDURE punitoth
IF printdes<>"s"
    ?
    ?
ENDIF (printdes<>"s")
IF PROW()>varpagel-9 .AND. printdes="p"
    DO WHILE PROW()<varpagel-2
        ?
    ENDDO (PROW()<varpagel-2)
    ? "PAGE "+LTRIM(STR(pagenova))
    pagenova=pagenova+1
    EJECT
ENDIF (PROW()>varpagel-9 .AND. printdes="p")
IF printdes="s" .AND. detailva="NO"
    CLEAR
ENDIF (printdes="s" .AND. detailva="NO")
? srcvar+" "+unitvar
? TRANSFORM(unitqty,"99")+" OF "+TRANSFORM(qtycount,"99")
? "PHASE "+phasevar
?? "   UNIT STRENGTH: "+TRANSFORM(m->strenvar,"999,999")
?
SELECT 24
GO TOP
*? "  A-Ration    B-Ration    T-Ration      MRE       LRP       R/CW
    WATER      BARRIER  CONSTRUCTION   MEDICAL"
?
DO WHILE .NOT. EOF()
    ?? TRANSFORM(class_i->unit_qty,"@R 999,999,999")+" "
                            288
```

```
        REPLACE class_i->unit_qty WITH 0
        SKIP
ENDDO (.NOT. EOF())
GO TOP
IF printdes="s"
        ?
ENDIF (printdes="s")
* Water
?? TRANSFORM(u_h2ovar, "@R 999,999,999")+" "
u_h2ovar=0
* CLASS IV
?? TRANSFORM(u_ivbvar, "@R 999,999,999")+" "
?? TRANSFORM(u_ivcvar, "@R 999,999,999")+" "
u_ivbvar=0
u_ivcvar=0
* Class 8
?? TRANSFORM(u_cl8var, "@R 999,999,999")
u_cl8var=0
SELECT 21
IF printdes="s" .AND. ROW()>20
        ?
        WAIT
        CLEAR
ENDIF (printdes="s" .AND. ROW()>20)
IF PROW()>varpagel-9 .AND. printdes="p"
        ?
        ? "PAGE "+LTRIM(STR(pagenova))
        pagenova=pagenova+1
        EJECT
ENDIF (PROW()>varpagel-9 .AND. printdes="p")
*!*****************************************************************
*!
*!        Procedure: PTASKOTH
*!
*!        Called by: PRIN_PRO                (procedure in SYSREP.PRG)
*!
*!*****************************************************************
PROCEDURE ptaskoth
IF PROW()>varpagel-15 .AND. printdes="p"
        DO WHILE PROW()<varpagel-2
            ?
        ENDDO (PROW()<varpagel-2)
        ? "PAGE "+LTRIM(STR(pagenova))
        pagenova=pagenova+1
        EJECT
ENDIF (PROW()>varpagel-15 .AND. printdes="p")
IF printdes="s".AND. ROW()>10 .AND. detailva="YES"
        WAIT
        CLEAR
ENDIF (printdes="s".AND. ROW()>10 .AND. detailva="YES")
****
IF printdes="s" .AND. detailva="NO"
        CLEAR
ENDIF (printdes="s" .AND. detailva="NO")
* List Task Org totals
?
?
? taskvar
? "TASK ORG  PHASE "+phasevar
?
```

289

```
*? "   A-Ration     B-Ration    T-Ration        MRE         LRP         R/CW
   WATER      BARRIER  CONSTRUCTION    MEDICAL"
?
SELECT 24
GO TOP
DO WHILE .NOT. EOF()
    ?? TRANSFORM(class_i->task_org_q,"@R 999,999,999")+" "
    REPLACE class_i->task_org_q WITH 0
    SKIP
ENDDO (.NOT. EOF())
GO TOP
IF printdes="s"
    ?
ENDIF (printdes="s")
* Water
?? TRANSFORM(t_h2ovar, "@R 999,999,999")+" "
t_h2ovar=0
* CLASS IV
?? TRANSFORM(t_ivbvar, "@R 999,999,999")+" "
?? TRANSFORM(t_ivcvar, "@R 999,999,999")+" "
t_ivbvar=0
t_ivcvar=0
* Class 8
?? TRANSFORM(t_cl8var, "@R 999,999,999")
t_cl8var=0
IF PROW()>varpagel-9 .AND. printdes="p"
    ?
    ? "PAGE "+LTRIM(STR(pagenova))
    pagenova=pagenova+1
    EJECT
ENDIF (PROW()>varpagel-9 .AND. printdes="p")
IF printdes="s"
    ?
    WAIT
    CLEAR
ENDIF (printdes="s")
*!****************************************************************
*!
*!      Procedure: PORDEOTH
*!
*!      Called by: PRIN_PRO          (procedure in SYSREP.PRG)
*!
*!****************************************************************
PROCEDURE pordeoth
?
?
IF PROW()>varpagel-15 .AND. printdes="p"
    DO WHILE PROW()<varpagel-2
        ?
    ENDDO (PROW()<varpagel-2)
    ? "PAGE "+LTRIM(STR(pagenova))
    pagenova=pagenova+1
    EJECT
ENDIF (PROW()>varpagel-15 .AND. printdes="p")
* List Order totals this phase
CLEAR
? "      ORDER OTHER CLASSES PHASE "+phasevar
?
SELECT 24
GO TOP
```

290

```
DO WHILE .NOT. EOF()
    ?? TRANSFORM(class_i->order_qty,"@R 999,999,999")+" "
    REPLACE class_i->order_qty WITH 0
    SKIP
ENDDO (.NOT. EOF())
GO TOP
SELECT 26
IF printdes="s"
    ?
ENDIF (printdes="s")
* Water
?? TRANSFORM(o_h2ovar, "@R 999,999,999")+" "
o_h2ovar=0
* CLASS IV
?? TRANSFORM(o_ivbvar, "@R 999,999,999")+" "
?? TRANSFORM(o_ivcvar, "@R 999,999,999")+" "
o_ivbvar=0
o_ivcvar=0
* Class 8
?? TRANSFORM(o_cl8var, "@R 999,999,999")
o_cl8var=0
IF PROW()>varpagel-9 .AND. printdes="p"
    ?
    ? "PAGE "+LTRIM(STR(pagenova))
    pagenova=pagenova+1
    EJECT
ENDIF (PROW()>varpagel-9 .AND. printdes="p")
IF printdes="s"
    ?
    WAIT
ENDIF (printdes="s")
CLEAR
*!******************************************************************
*!
*!       Procedure: REPORTHD
*!
*!       Called by: PRIN_PRO          (procedure in SYSREP.PRG)
*!
*!******************************************************************
PROCEDURE reporthd
IF printdes<>"s"
    ?
    ?
ENDIF (printdes<>"s")
IF PROW()>varpagel-4 .AND. printdes="p"
    DO WHILE PROW()<varpagel-2
        ?
    ENDDO (PROW()<varpagel-2)
    ? "PAGE "+LTRIM(STR(pagenova))
    pagenova=pagenova+1
    EJECT
ENDIF (PROW()>varpagel-4 .AND. printdes="p")
IF printdes="s".AND. ROW()>18
    WAIT
    CLEAR
ENDIF (printdes="s".AND. ROW()>18)
* Need to choose phase length
plenvar="PHASE"+LTRIM(STR(cl_other->phase))+"LEN"
? "ORDER: "+cl_other->number+" "+orderlst->name
?
```

```
IF vttlva<>"o"
    ? "TASK ORGANIZATION: "+cl_other->taskorgnam
ENDIF (vttlva<>"o")
phasevar=LTRIM(STR(cl_other->phase))
? "Phase: "+phasevar+"  Strength "+LTRIM(STR(ordertas->strength))+"%"
?? "  Phase Length: "
DO CASE
CASE plenvar="PHASE1LEN"
    ?? LTRIM(STR(orderlst->phase1len))
CASE plenvar="PHASE2LEN"
    ?? LTRIM(STR(orderlst->phase2len))
CASE plenvar="PHASE3LEN"
    ?? LTRIM(STR(orderlst->phase3len))
CASE plenvar="PHASE4LEN"
    ?? LTRIM(STR(orderlst->phase4len))
CASE plenvar="PHASE5LEN"
    ?? LTRIM(STR(orderlst->phase5len))
CASE plenvar="PHASE6LEN"
    ?? LTRIM(STR(orderlst->phase6len))
CASE plenvar="PHASE7LEN"
    ?? LTRIM(STR(orderlst->phase7len))
CASE plenvar="PHASE8LEN"
    ?? LTRIM(STR(orderlst->phase8len))
CASE plenvar="PHASE9LEN"
    ?? LTRIM(STR(orderlst->phase9len))
ENDCASE
? "Parameter Set: "+ordertas->profilenam+"  "+cprofile->profiledes
? "       Profile: "+cprofile->profile+"  "
?? "  Class VIII: "+cprofile->med_name+"   "
? "         Water: "+cprofile->water_name
?? " Ration Cycle: "+cprofile->cl_i_1+" "+cprofile->cl_i_2+" "+cprofile->cl_i_3
?
? "   A-Ration     B-Ration     T-Ration        MRE         LRP       R/CW"
IF printdes="s"
    ?
ENDIF (printdes="s")
?? "        WATER     BARRIER  CONSTRUCTION   MEDICAL"
*!*******************************************************************
*!
*!      Procedure: DETA_PRO
*!
*!      Called by: PRINTPOL.PRG
*!               : PRINTAMM.PRG
*!               : PRINTOTH.PRG
*!
*!*******************************************************************
PROCEDURE deta_pro
DO CASE
CASE BAR()=1
   vttlva="uto"
   DEACTIVATE POPUP
CASE BAR()=2
   vttlva="to"
   DEACTIVATE POPUP
CASE BAR()=3
   vttlva="o"
   DEACTIVATE POPUP
ENDCASE
CLEAR
* Bar 1 OF deta_pop PROMPT "Unit, Task Org, and Order Totals"
```

```
*!*********************************************************************
*!
*!        Procedure: DETA_BAR1
*!
*!*********************************************************************
PROCEDURE deta_bar1
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
    Consumption data is printed for water, rations, barrier and
    construction material, and medical supplies (by weight) for
    each Unit, Task Organization by Phase, Order by Phase and Order
    Total.
ENDTEXT

RETURN
* BAR 2 OF deta_pop PROMPT "Task Org and Order Totals Only"
*!*********************************************************************
*!
*!        Procedure: DETA_BAR2
*!
*!*********************************************************************
PROCEDURE deta_bar2
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
    Consumption data is printed for water, rations, barrier and
    construction material, and medical supplies (by weight) for
    each Task Organization by Phase, Order by Phase and Order
    Total.
ENDTEXT

RETURN
* BAR 3 OF deta_pop PROMPT "Task Org and Order Totals Only"
*!*********************************************************************
*!
*!        Procedure: DETA_BAR3
*!
*!*********************************************************************
PROCEDURE deta_bar3
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
    Consumption data is printed for water, rations, barrier and
    construction material, and medical supplies (by weight) for
    each Order by Phase and for the Order Total.
ENDTEXT

RETURN
* BAR 1 OF prin_pop PROMPT "Print Estimate to Printer"
*!*********************************************************************
*!
*!        Procedure: PRIN_BAR1
*!
*!*********************************************************************
PROCEDURE prin_bar1
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
    Make sure printer is on and on-line.
```

```
ENDTEXT

RETURN
* BAR 2 OF prin_pop PROMPT "Print Estimate to Screen"
*!********************************************************************
*!
*!        Procedure: PRIN_BAR2
*!
*!********************************************************************
PROCEDURE prin_bar2
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
    Prints Reports to the Screen. These estimates can take a long time
    to view this way.  It may be preferred to print to a file and then
    view the file with a work processor.
ENDTEXT

RETURN
* BAR 3 OF prin_pop PROMPT "Print Estimate to ASCII File"
*!********************************************************************
*!
*!        Procedure: PRIN_BAR3
*!
*!********************************************************************
PROCEDURE prin_bar3
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
    ASCII files can be viewed and edited with most word processors, or
    with EDIT if you are using a version of DOS 5.0 or higher.
ENDTEXT

RETURN
* BAR 4 OF prin_pop PROMPT "Change Printer Defaults"
*!********************************************************************
*!
*!        Procedure: PRIN_BAR4
*!
*!********************************************************************
PROCEDURE prin_bar4
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
    Change printer page length and left margin.
ENDTEXT

RETURN
* BAR 5 OF prin_pop PROMPT "Return to Previous Menu"
*!********************************************************************
*!
*!        Procedure: PRIN_BAR5
*!
*!********************************************************************
PROCEDURE prin_bar5
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT

ENDTEXT
```

RETURN

*: EOF: PRINTOTH.PRG

```
*:*******************************************************************
*:            '
*: Procedure file: D:\SURE\PRINTPOL.PRG
*:
*:    Procs & Fncts: STARTPLR
*:                  : DETA_PRO
*:                  : POLCALC
*:                  : PUNITPOL
*:                  : PTASKPOL
*:                  : PORDEPOL
*:                  : LINPOL
*:                  : REPORTHD
*:                  : TFHEAD
*:                  : DETA_BAR1
*:                  : DETA_BAR2
*:                  : DETA_BAR3
*:                  : DETA_BAR4
*:                  : PRIN_BAR1
*:                  : PRIN_BAR2
*:                  : PRIN_BAR3
*:                  : PRIN_BAR4
*:                  : PRIN_BAR5
*:
*:          Set by: ORDR_PRO              (procedure in ORDEREP1.PRG)
*:
*:          Calls: STARTPLR              (procedure in PRINTPOL.PRG)
*:               : DETA_PRO              (procedure in PRINTPOL.PRG)
*:               : PRIN_PRO              (procedure in SYSREP.PRG)
*:
*:        Documented 04/16/94 at 11:02              SNAP!  version 5.00
*:*******************************************************************
STORE PROGRAM() TO progvar
STORE SPACE(60) TO filevar
STORE " NO" TO detailva
STORE " " TO printdes
STORE SPACE(3) TO vttlva
STORE SPACE(9) TO plenvar
STORE 1 TO unitqty
STORE 1 TO qtycount
CLEAR
* Setup Printers for Condensed Print
DO CASE
CASE vprintva="Dot Matrix"
   ??? CHR(27)+CHR(15)
CASE vprintva="Laser"
CASE vprintva="DeskJet"
   ??? CHR(27)+"(s16.66H"
ENDCASE
DO startplr
DEFINE POPUP deta_pop FROM 1,2
DEFINE BAR 1 OF deta_pop PROMPT "LIN by LIN Detail, plus all Totals"
DEFINE BAR 2 OF deta_pop PROMPT "Summary Report - Unit, Task Org, and Order
Totals"
DEFINE BAR 3 OF deta_pop PROMPT "Summary Report - Task Org and Order Totals
Only"
DEFINE BAR 4 OF deta_pop PROMPT "Summary Report - Order Totals Only"
ON SELECTION POPUP deta_pop DO deta_pro
ON BAR 1 OF deta_pop DO deta_bar1
ON BAR 2 OF deta_pop DO deta_bar2
ON BAR 3 OF deta_pop DO deta_bar3
```

296

```
ON BAR 4 OF deta_pop DO deta_bar4
DEFINE POPUP prin_pop FROM 1,1
DEFINE BAR 1 OF prin_pop PROMPT "Print Estimate to Printer"
DEFINE BAR 2 OF prin_pop PROMPT "Print Estimate to Screen"
DEFINE BAR 3 OF prin_pop PROMPT "Print Estimate to ASCII File"
DEFINE BAR 4 OF prin_pop PROMPT "Change Printer Defaults"
DEFINE BAR 5 OF prin_pop PROMPT "Return to Previous Menu"
ON SELECTION POPUP prin_pop DO prin_pro
ON BAR 1 OF prin_pop DO prin_bar1
ON BAR 2 OF prin_pop DO prin_bar2
ON BAR 3 OF prin_pop DO prin_bar3
ON BAR 4 OF prin_pop DO prin_bar4
ON BAR 5 OF prin_pop DO prin_bar5
DO WHILE PROMPT()<>"Return to Previous Menu"
    ACTIVATE POPUP prin_pop
ENDDO (PROMPT()<>"Return to Previous Menu")
*!*******************************************************************
*!
*!        Procedure: PRIN_PRO
*!
*!        Called by: SYSREP.PRG
*!                 : PRINTPOL.PRG
*!                 : PRINTAMM.PRG
*!                 : PRINTOTH.PRG
*!
*!           Calls: PRNSETUP.PRG
*!                 : STARTPLR            (procedure in PRINTPOL.PRG)
*!                 : REPORTHD            (procedure in PRINTPOL.PRG)
*!                 : TFHEAD              (procedure in PRINTPOL.PRG)
*!                 : LINPOL              (procedure in PRINTPOL.PRG)
*!                 : POLCALC             (procedure in PRINTPOL.PRG)
*!                 : PUNITPOL            (procedure in PRINTPOL.PRG)
*!                 : PTASKPOL            (procedure in PRINTPOL.PRG)
*!                 : PORDEPOL            (procedure in PRINTPOL.PRG)
*!                 : LINAMM              (procedure in PRINTAMM.PRG)
*!                 : AMMOCALC            (procedure in PRINTAMM.PRG)
*!                 : PUNITAMM            (procedure in PRINTAMM.PRG)
*!                 : PTASKAMM            (procedure in PRINTAMM.PRG)
*!                 : PORDEAMM            (procedure in PRINTAMM.PRG)
*!                 : OTHECALC            (procedure in PRINTOTH.PRG)
*!                 : PUNITOTH            (procedure in PRINTOTH.PRG)
*!                 : PTASKOTH            (procedure in PRINTOTH.PRG)
*!                 : PORDEOTH            (procedure in PRINTOTH.PRG)
*!
*!*******************************************************************
PROCEDURE prin_pro
DO CASE
CASE BAR()=1
    printdes="p"
CASE BAR()=2
    printdes="s"
    CLEAR
CASE BAR()=3
    printdes="f"
    ACTIVATE WINDOW process
    @ 1,1 SAY "Enter path and file name."
    @ 3,1 GET filevar
    READ
    SET ALTERNATE TO &filevar
    DEACTIVATE WINDOW process
```

```
CASE BAR()=4'
   DO prnsetup
CASE BAR()=5
   DEACTIVATE POPUP
   RETURN
ENDCASE
CLEAR
ACTIVATE POPUP deta_pop
* Set according to destination
DO CASE
CASE printdes="p"
   SET DEVICE TO PRINT
   SET PRINT ON
CASE printdes="s"
CASE printdes="f"
   SET ALTERNATE ON
ENDCASE
* Initialize Variables
STORE " " TO phasevar
taskvar=SPACE(60)
STORE 1 TO pagenova
***********These variable are hard coded. May be replaced with array. Memory?
* track Unit JP4 consumption this phase
STORE 0 TO jp4
* track Unit JP4 consumption total
STORE 0 TO jp4t
* track Task Org JP4 consumption this phase
STORE 0 TO jp4run
* track Task Org JP4 consumption total
STORE 0 TO jp4runt
* track Order JP4 consumption this phase
STORE 0 TO jp4con
* track Order JP4 consumption total
STORE 0 TO jp4cont
* track Unit JP4 aircraft storage
STORE 0 TO jp4s
* track Task Org JP4 aircraft tank storage
STORE 0 TO jp4tnkr
* track Order JP4 aircraft tank storage
STORE 0 TO jp4tnkt
* track Unit DSL consumption this phase
STORE 0 TO dsl
* track Unit DSL consumption total
STORE 0 TO dslt
* track Task Org DSL consumption this phase
STORE 0 TO dslrun
* track Task Org DSL consumption total
STORE 0 TO dslrunt
* track Order DSL consumption this phase
STORE 0 TO dslcon
* track Order DSL consumption total
STORE 0 TO dslcont
* track Unit DSL tank storage
STORE 0 TO dsls
* track Task Org DSL tank storage
STORE 0 TO dsltnkr
* track Order DSL tank storage
STORE 0 TO dsltnkt
* track Unit MOG consumption this phase
STORE 0 TO mog
```

```
* track Unit MOG consumption total
STORE 0 TO mogt
* track Task Org MOG consumption this phase
STORE 0 TO mogrun
* track Task Org MOG consumption total
STORE 0 TO mogrunt
* track Order MOG consumption this phase
STORE 0 TO mogcon
* track Order MOG consumption total
STORE 0 TO mogcont
* track Unit MOG tank storage
STORE 0 TO mogs
* track Task Org MOG tank storage
STORE 0 TO mogtnkr
* track Order MOG tank storage
STORE 0 TO mogtnkt
* Will need to track each unit as done
STORE SPACE(60) TO unitvar
STORE 0 TO ordervar
STORE 0 TO countvar
STORE SPACE(60) TO taskvar
* dbf's and variables
DO startplr
SELECT 20
SET ORDER TO unitname
ordervar=NUMBER
IF printdes="f" .OR. printdes="p"
    ACTIVATE WINDOW process
ENDIF (printdes="f" .OR. printdes="p")
countvar=1
SELECT 18
SET FILTER TO
SET FILTER TO NUMBER=ordervar
SET ORDER TO phasetas
GO TOP
IF EOF()
    ACTIVATE WINDOW small
    WAIT "ERROR"
    DEACTIVATE WINDOW small
    QUIT
ENDIF (EOF())
CLEAR
DO WHILE .T.                    && ordertas
    SELECT 20
    SET FILTER TO
    GO TOP
    SET FILTER TO taskorgnam=ordertas->taskorgnam .AND. phase=ordertas->phase
    GO TOP
    ****
    IF printdes="s" .AND. detailva="NO"
        CLEAR
    ENDIF (printdes="s" .AND. detailva="NO")
    DO reporthd
    IF PROW()>varpagel-9 .AND. printdes="p"
        ? "PAGE "+LTRIM(STR(pagenova))
        pagenova=pagenova+1
        EJECT
    ENDIF (PROW()>varpagel-9 .AND. printdes="p")
    IF printdes="s"                && .AND. row()>20
        ?
```

299

```
    WAIT
    CLEAR
ENDIF (printdes="s"                    &&.AND. row()>20)
?
*    ?
*    ? taskorgnam
*    ?"_____ "
DO tfhead
?
taskvar=polest->taskorgnam
STORE RECNO() TO recnovar
qtycount=polest->qty
DO WHILE unitqty<=qtycount        &&polest->qty
    unitvar=polest->unitname
    *qtycount=polest->qty
    IF detailva="YES"
        ? src+" "+unitvar
        ? TRANSFORM(unitqty,"99")+" OF "+TRANSFORM(qtycount,"99")
        IF printdes<>"s"
            ? "LIN    NOMENCLATURE                     QTY EC    POL    CONSUMPTION
TANK CAPACITY              WEIGHT"
            ? "---    ------------                     --- --    ---    -----------
-------------      ------"
        ELSE
            ? "LIN    NOMENCLATURE                     QTY EC    POL    CONSUMPTION
TANK CAPACITY"
            ? "---    ------------                     --- --    ---    -----------
-------------"
        ENDIF (printdes<>"s")
    ENDIF (detailva="YES")
    ****
    IF printdes="s" .AND. detailva="NO"
        CLEAR
    ENDIF (printdes="s" .AND. detailva="NO")
    DO WHILE .T.       &&.NOT. EOF()
        DO linpol
        DO polcalc
        SKIP
        IF unitname<>unitvar
            DO punitpol
            EXIT
        ENDIF (unitname<>unitvar)
    ENDDO (.T.        &&.NOT. EOF())
    IF unitqty<qtycount
        unitqty=unitqty+1
        GO recnovar
        IF printdes="s" .AND. detailva="NO"
            CLEAR
        ENDIF (printdes="s" .AND. detailva="NO")
        LOOP
    ELSE
        IF .NOT. EOF()
            STORE RECNO() TO recnovar
            unitvar=polest->unitname
            unitqty=1
            qtycount=polest->qty
            IF printdes="s" .AND. detailva="NO"
                CLEAR
            ENDIF (printdes="s" .AND. detailva="NO")
        ELSE
```

300

```
            EXIT
          ENDIF (.NOT. EOF())
       ENDIF (unitqty<qtycount)
     ENDDO (unitqty<=qtycount      &&polest->qty)
     unitqty=1
     DO ptaskpol
     SELECT 18      && ordertas
     SKIP
     IF LTRIM(STR(phase))<>phasevar
        DO pordepol
     ENDIF (LTRIM(STR(phase))<>phasevar)
     STORE LTRIM(STR(phase))TO phasevar
     IF EOF()
        EXIT
     ENDIF (EOF())
ENDDO                  && ordertas (.T.              && ordertas) (.T.
      && ordertas)
* List Order totals
?
?
IF printdes<>"s"
   ? "ORDER FUEL TOTAL                              CONSUMPTION
               WEIGHT"
   **NOTE: SOME HARD CODING OF FUEL WEIGHTS IN FOLLOWING!
   ? "                                        JP4:
"+TRANSFORM(jp4cont,"999,999,999.99")+"
"+TRANSFORM(jp4cont*6.375,"9,999,999,999.99")
   ? "                                        DSL:
"+TRANSFORM(dslcont,"999,999,999.99")+"
"+TRANSFORM(dslcont*7.034,"9,999,999,999.99")
   ? "                                        MOG:
"+TRANSFORM(mogcont,"999,999,999.99")+"
"+TRANSFORM(mogcont*6.203,"9,999,999,999.99")
   ?
ELSE
   ? "ORDER FUEL TOTAL CONSUMPTION              WEIGHT"
   **NOTE: SOME HARD CODING OF FUEL WEIGHTS IN FOLLOWING!
   ? "         JP4: "+TRANSFORM(jp4cont,"999,999,999.99")+"
"+TRANSFORM(jp4cont*6.375,"9,999,999,999.99")
   ? "         DSL: "+TRANSFORM(dslcont,"999,999,999.99")+"
"+TRANSFORM(dslcont*7.034,"9,999,999,999.99")
   ? "         MOG: "+TRANSFORM(mogcont,"999,999,999.99")+"
"+TRANSFORM(mogcont*6.203,"9,999,999,999.99")
   ?
ENDIF (printdes<>"s")
* Print last page number
IF PROW()>1 .AND. printdes="p"
   DO WHILE PROW()<varpagel-1
      ?
   ENDDO (PROW()<varpagel-1)
   ? "PAGE "+LTRIM(STR(pagenova))
ENDIF (PROW()>1 .AND. printdes="p")
SET PRINT OFF
SET DEVICE TO SCREEN
IF printdes="s"
   ?
   WAIT
   CLEAR
ENDIF (printdes="s")
IF printdes="f"
```

```
    ?
    ? "END OF REPORT"
    SET ALTERNATE OFF
    SET DEVICE TO SCREEN
    DEACTIVATE WINDOW process
ENDIF (printdes="f")
IF printdes="p"
    DEACTIVATE WINDOW process
ENDIF (printdes="p")
SELECT 18
SET FILTER TO
SELECT 20
SET FILTER TO
*!*****************************************************************
*!
*!          Procedure: STARTPLR
*!
*!          Called by: PRINTPOL.PRG
*!                   : PRINTAMM.PRG
*!                   : PRINTOTH.PRG
*!                   : PRIN_PRO               (procedure in SYSREP.PRG)
*!
*!              Calls: STARTHLP.PRG
*!
*!               Uses: TASKORG.DBF
*!                   : CSRCLIST.DBF
*!                   : ORDERLST.DBF
*!                   : CPROFILE.DBF
*!                   : PROFILES.DBF
*!                   : POLRATE.DBF
*!                   : POSTURE.DBF
*!                   : INTENSTY.DBF
*!                   : POLWGT.DBF
*!                   : CSRCLINQ.DBF
*!                   : ORDERTAS.DBF
*!                   : DODIC.DBF
*!                   : POLEST.DBF
*!                   : AMMOEST.DBF
*!                   : AMMORATE.DBF
*!                   : MEDICAL.DBF
*!                   : WATER.DBF
*!                   : CLASS_I.DBF
*!                   : CL_OTHER.DBF
*!
*!          MDX files: TASKORG.MDX
*!                   : CSRCLIST.MDX
*!                   : ORDERLST.MDX
*!                   : CPROFILE.MDX
*!                   : PROFILES.MDX
*!                   : POLRATE.MDX
*!                   : INTENSTY.MDX
*!                   : POLWGT.MDX
*!                   : CSRCLINQ.MDX
*!                   : ORDERTAS.MDX
*!                   : DODIC.MDX
*!                   : POLEST.MDX
*!                   : AMMOEST.MDX
*!                   : AMMORATE.MDX
*!                   : MEDICAL.MDX
*!                   : WATER.MDX
```

```
*!                      : CLASS_I.MDX
*!                      : CL_OTHER.MDX
*!
*!***************************************************************
PROCEDURE startplr
* OPEN FILES
CLOSE DATA
DO starthlp
STORE 1 TO statusvar
SELECT 1
USE taskorg
SELECT 2
USE csrclist
SELECT 3
USE orderlst
SELECT 4
USE cprofile ORDER profilenam
SELECT 7
USE profiles ORDER profile
*SELECT 8
*USE ammorate ORDER lin
SELECT 9
USE polrate
SELECT 17
USE posture
SELECT 10
USE intensty
SELECT 12
USE polwgt ORDER pol_type
SELECT 14
USE csrclinq
SELECT 18
USE ordertas
SELECT 19
USE dodic ORDER dodic
SELECT 20
USE polest
SELECT 21
USE ammoest
* SET RELATIONS
SELECT 4
SET ORDER TO profile
SET RELATION TO profile INTO profiles
SET ORDER TO profilenam
SELECT 18
SET RELATION TO profilenam INTO cprofile
SELECT 3
USE orderlst
SET ORDER TO NUMBER
SEEK polest->number
SELECT 14
SET ORDER TO lin
SELECT 20
SET ORDER TO lin
SET RELATION TO lin INTO csrclinq
*!***************************************************************
*!
*!        Procedure: POLCALC
*!
*!        Called by: CALCPOL.PRG
```
303

```
*!                    : PRIN_PRO            (procedure in SYSREP.PRG)
*!
*!***********************************************************************
PROCEDURE polcalc
DO CASE
CASE pol_type="JP4"
   * track Unit JP4 consumption this phase
   jp4=jp4+pol_qty
   * track Unit JP4 consumption total
   jp4t=jp4t+pol_qty
   * track Task Org JP4 consumption this phase
   jp4run=jp4run+pol_qty
   * track Tsk Org JP4 consumption total
   jp4runt=jp4runt+pol_qty
   * track Order JP4 consumption this phase
   jp4con=jp4con+pol_qty
   * track Order JP4 consumption total
   jp4cont=jp4cont+pol_qty
   * track Unit aircraft tank storage
   jp4s=jp4s+ft_cap
   * track Task Org JP4 aircraft tank storage
   jp4tnkr=jp4tnkr+ft_cap
   * track Order JP4 aircraft tank storage
   jp4tnkt=jp4tnkt+ft_cap
CASE pol_type="DSL"
   * track Unit DSL consumption this phase
   dsl=dsl+pol_qty
   * track Unit DSL consumption total
   dslt=dslt+pol_qty
   * track Task Org DSL consumption this phase
   dslrun=dslrun+pol_qty
   * track Task Org DSL consumption total
   dslrunt=dslrunt+pol_qty
   * track Order DSL consumption this phase
   dslcon=dslcon+pol_qty
   * track Order DSL consumption total
   dslcont=dslcont+pol_qty
   *track Unit DSL tank storage
   dsls=dsls+ft_cap
   * track Task Org DSL tank storage
   dsltnkr=dsltnkr+ft_cap
   * track Order DSL aircraft tank storage
   dsltnkt=dsltnkt+ft_cap
CASE pol_type="MOG"
   * track Unit MOG consumption this phase
   mog=mog+pol_qty
   * track Unit MOG consumption total
   mogt=mogt+pol_qty
   * track Task Org MOG consumption this phase
   mogrun=mogrun+pol_qty
   * track Task Org MOG consumption total
   mogrunt=mogrunt+pol_qty
   * track Order MOG consumption this phase
   mogcon=mogcon+pol_qty
   * track Order MOG consumption total
   mogcont=mogcont+pol_qty
   * track Unit MOG tank storage
   mogs=mogs+ft_cap
   * track Task Org MOG tank storage
   mogtnkr=mogtnkr+ft_cap
```

304

```
      * track Order MOG tank storage
      mogtnkt=mogtnkt+ft_cap
ENDCASE
*!***********************************************************************
*!
*!       Procedure: PUNITPOL
*!
*!       Called by: PRIN_PRO          (procedure in SYSREP.PRG)
*!
*!***********************************************************************
PROCEDURE punitpol
IF vttlva<>"to" .AND. vttlva<>"o"
      ?
      ?
      IF PROW()>varpagel-10 .AND. printdes="p"
         ? "PAGE "+LTRIM(STR(pagenova))
         pagenova=pagenova+1
         EJECT
      ENDIF (PROW()>varpagel-10 .AND. printdes="p")
      IF printdes="s" .AND. detailva="YES"
         WAIT
         CLEAR
      ENDIF (printdes="s" .AND. detailva="YES")
      ****
      IF printdes="s" .AND. detailva="NO"
         CLEAR
      ENDIF (printdes="s" .AND. detailva="NO")
      ? unitvar
      ? TRANSFORM(unitqty,"99")+" OF "+TRANSFORM(qtycount,"99")
      IF printdes<>"s"
         ? "UNIT FUEL FOR PHASE "+phasevar+"
CONSUMPTION        TANK CAPACITY              WEIGHT"
         **NOTE: SOME HARD CODING OF FUEL WEIGHTS IN FOLLOWING!
         ? "                                          JP4:
"+TRANSFORM(jp4,"999,999,999.99")+"      "+TRANSFORM(jp4s,"999,999,999.99")+"
"+TRANSFORM(jp4*6.375,"9,999,999,999.99")
         ? "                                          DSL:
"+TRANSFORM(dsl,"999,999,999.99")+"      "+TRANSFORM(dsls,"999,999,999.99")+"
"+TRANSFORM(dsl*7.034,"9,999,999,999.99")
         ? "                                          MOG:
"+TRANSFORM(mog,"999,999,999.99")+"      "+TRANSFORM(mogs,"999,999,999.99")+"
"+TRANSFORM(mog*6.203,"9,999,999,999.99")
      ELSE
         ? "UNIT FUEL FOR PHASE "+phasevar+"        CONSUMPTION        TANK CAPACITY
            WEIGHT"
         **NOTE: SOME HARD CODING OF FUEL WEIGHTS IN FOLLOWING!
         ? "                    JP4: "+TRANSFORM(jp4,"999,999,999.99")+"
"+TRANSFORM(jp4s,"999,999,999.99")+"  "+TRANSFORM(jp4*6.375,"9,999,999,999.99")
         ? "                    DSL: "+TRANSFORM(dsl,"999,999,999.99")+"
"+TRANSFORM(dsls,"999,999,999.99")+"  "+TRANSFORM(dsl*7.034,"9,999,999,999.99")
         ? "                    MOG: "+TRANSFORM(mog,"999,999,999.99")+"
"+TRANSFORM(mogs,"999,999,999.99")+"  "+TRANSFORM(mog*6.203,"9,999,999,999.99")
      ENDIF (printdes<>"s")
      ?
      IF printdes="s"
         ?
         WAIT
         CLEAR
      ENDIF (printdes="s")
ENDIF (vttlva<>"to" .AND. vttlva<>"o")
```

```
*Reset Unit variables
jp4=0
dsl=0
mog=0
jp4s=0
dsls=0
mogs=0
*!******************************************************************
*!
*!         Procedure: PTASKPOL
*!
*!         Called by: PRIN_PRO            (procedure in SYSREP.PRG)
*!
*!******************************************************************
PROCEDURE ptaskpol
IF vttlva<>"o"
    ?
    ?
    IF PROW()>varpagel-9 .AND. printdes="p"
       ? "PAGE "+LTRIM(STR(pagenova))
       pagenova=pagenova+1
       EJECT
    ENDIF (PROW()>varpagel-9 .AND. printdes="p")
    IF printdes="s".AND. ROW()>20 .AND. detailva="YES"
       WAIT
       CLEAR
    ENDIF (printdes="s".AND. ROW()>20 .AND. detailva="YES")
    ****
    IF printdes="s" .AND. detailva="NO"
       CLEAR
    ENDIF (printdes="s" .AND. detailva="NO")
    * List Task Org totals
    ? taskvar
    ?
    IF printdes<>"s"
       ? "TASK ORG FUEL FOR PHASE "+phasevar+"
CONSUMPTION     TANK CAPACITY              WEIGHT"
       **NOTE: SOME HARD CODING OF FUEL WEIGHTS IN FOLLOWING!
       ? "                                             JP4:
"+TRANSFORM(jp4run,"999,999,999.99")+"
"+TRANSFORM(jp4tnkr,"999,999,999.99")+"
"+TRANSFORM(jp4run*6.375,"9,999,999,999.99")
       ? "                                             DSL:
"+TRANSFORM(dslrun,"999,999,999.99")+"
"+TRANSFORM(dsltnkr,"999,999,999.99")+"
"+TRANSFORM(dslrun*7.034,"9,999,999,999.99")
       ? "                                             MOG:
"+TRANSFORM(mogrun,"999,999,999.99")+"
"+TRANSFORM(mogtnkr,"999,999,999.99")+"
"+TRANSFORM(mogrun*6.203,"9,999,999,999.99")
    ELSE
       ? "TASK ORG FUEL FOR PHASE "+phasevar+" CONSUMPTION      TANK CAPACITY
         WEIGHT"
       **NOTE: SOME HARD CODING OF FUEL WEIGHTS IN FOLLOWING!
       ? "                      JP4: "+TRANSFORM(jp4run,"999,999,999.99")+"
"+TRANSFORM(jp4tnkr,"999,999,999.99")+"
"+TRANSFORM(jp4run*6.375,"9,999,999,999.99")
       ? "                      DSL: "+TRANSFORM(dslrun,"999,999,999.99")+"
"+TRANSFORM(dsltnkr,"999,999,999.99")+"
"+TRANSFORM(dslrun*7.034,"9,999,999,999.99")
```

```
       ? "                       MOG: "+TRANSFORM(mogrun,"999,999,999.99")+"
"+TRANSFORM(mogtnkr,"999,999,999.99")+"
"+TRANSFORM(mogrun*6.203,"9,999,999,999.99")
    ENDIF (printdes<>"s")
    ?
ENDIF (vttlva<>"o")
* Reset Task Org variables
jp4run=0
dslrun=0
mogrun=0
jp4tnkr=0
dsltnkr=0
mogtnkr=0
IF printdes="s"
    ?
    WAIT
    CLEAR
ENDIF (printdes="s")
*!****************************************************************************
*!
*!      Procedure: PORDEPOL
*!
*!      Called by: PRIN_PRO              (procedure in SYSREP.PRG)
*!
*!****************************************************************************
PROCEDURE pordepol
?
?
IF PROW()>varpagel-9 .AND. printdes="p"
    ? "PAGE "+LTRIM(STR(pagenova))
    pagenova=pagenova+1
    EJECT
ENDIF (PROW()>varpagel-9 .AND. printdes="p")
*IF printdes="s" &&.AND. row()>14
*    WAIT
*    CLEAR
*ENDIF
* List Order totals this phase
IF printdes<>"s"
    ? "ORDER FUEL FOR PHASE "+phasevar+"                          CONSUMPTION
    TANK CAPACITY           WEIGHT"
    **NOTE: SOME HARD CODING OF FUEL WEIGHTS IN FOLLOWING!
    ? "                                       JP4:
"+TRANSFORM(jp4con,"999,999,999.99")+"
"+TRANSFORM(jp4tnkt,"999,999,999.99")+"
"+TRANSFORM(jp4con*6.375,"9,999,999,999.99")
    ? "                                       DSL:
"+TRANSFORM(dslcon,"999,999,999.99")+"
"+TRANSFORM(dsltnkt,"999,999,999.99")+"
"+TRANSFORM(dslcon*7.034,"9,999,999,999.99")
    ? "                                       MOG:
"+TRANSFORM(mogcon,"999,999,999.99")+"
"+TRANSFORM(mogtnkt,"999,999,999.99")+"
"+TRANSFORM(mogcon*6.203,"9,999,999,999.99")
ELSE
    ? "ORDER FUEL FOR PHASE "+phasevar+"   CONSUMPTION    TANK CAPACITY
    WEIGHT"
    **NOTE: SOME HARD CODING OF FUEL WEIGHTS IN FOLLOWING!
```

```
    ? "                         JP4: "+TRANSFORM(jp4con,"999,999,999.99")+"
"+TRANSFORM(jp4tnkt,"999,999,999.99")+"
"+TRANSFORM(jp4con*6.375,"9,999,999,999.99")
    ? "                         DSL: "+TRANSFORM(dslcon,"999,999,999.99")+"
"+TRANSFORM(dsltnkt,"999,999,999.99")+"
"+TRANSFORM(dslcon*7.034,"9,999,999,999.99")
    ? "                         MOG: "+TRANSFORM(mogcon,"999,999,999.99")+"
"+TRANSFORM(mogtnkt,"999,999,999.99")+"
"+TRANSFORM(mogcon*6.203,"9,999,999,999.99")
ENDIF (printdes<>"s")
?
IF printdes="s"
    ?
    WAIT
    CLEAR
ENDIF (printdes="s")
* Reset Order Fuel variables
jp4con=0
dslcon=0
mogcon=0
jp4tnkt=0
dsltnkt=0
mogtnkt=0
*!*********************************************************************
*!
*!        Procedure: LINPOL
*!
*!        Called by: PRIN_PRO              (procedure in SYSREP.PRG)
*!
*!*********************************************************************
PROCEDURE linpol
* Do one LIN at a time
IF detailva="YES"
.    STORE TRANSFORM(pol_qty,"@R 999,999,999.99") TO pol1var
    STORE TRANSFORM(ft_cap,"@R 999,999,999.99") TO pol2var
    STORE TRANSFORM(pol_qty*lbs_gal,"@R 999,999,999.99") TO pol3var
    ? lin+" "+SUBSTR(lin_nomen,1,25)+" "+TRANSFORM(quantity,"9999")+"
"+equip_cat+" ";
        +" "+pol_type+" "+pol1var+"        "+pol2var
    IF printdes<>"s"
        ??"        "+pol3var
    ENDIF (printdes<>"s")
ENDIF (detailva="YES")
IF PROW()>varpagel-5 .AND. printdes="p"
    ? "PAGE "+LTRIM(STR(pagenova))
    pagenova=pagenova+1
    EJECT
ENDIF (PROW()>varpagel-5 .AND. printdes="p")
IF printdes="s" .AND. ROW()>20
    WAIT
    CLEAR
ENDIF (printdes="s" .AND. ROW()>20)
*!*********************************************************************
*!
*!        Procedure: REPORTHD
*!
*!        Called by: PRIN_PRO              (procedure in SYSREP.PRG)
*!
*!*********************************************************************
PROCEDURE reporthd
```

308

```
?
?
IF PROW()>varpagel-4 .AND. printdes="p"
    ? "PAGE "+LTRIM(STR(pagenova))
    pagenova=pagenova+1
    EJECT
ENDIF (PROW()>varpagel-4 .AND. printdes="p")
IF printdes="s".AND. ROW()>18
    WAIT
    CLEAR
ENDIF (printdes="s".AND. ROW()>18)
* Need to choose phase length
plenvar="PHASE"+LTRIM(STR(polest->phase))+"LEN"
? polest->number+" "+orderlst->name
IF printdes="s"
    ?
ENDIF (printdes="s")
??"  ESTIMATED BULK POL CONSUMPTION REPORT as of "+DTOC(DATE())
*!********************************************************************
*!
*!       Procedure: TFHEAD
*!
*!       Called by: PRIN_PRO              (procedure in SYSREP.PRG)
*!
*!********************************************************************
PROCEDURE tfhead
? polest->taskorgnam
phasevar=LTRIM(STR(polest->phase))
? "Phase: "+phasevar+"  Strength "+LTRIM(STR(ordertas->strength))+"%"
?? "  Phase Length: "
DO CASE
CASE plenvar="PHASE1LEN"
    ?? LTRIM(STR(orderlst->phase1len))
CASE plenvar="PHASE2LEN"
    ?? LTRIM(STR(orderlst->phase2len))
CASE plenvar="PHASE3LEN"
    ?? LTRIM(STR(orderlst->phase3len))
CASE plenvar="PHASE4LEN"
    ?? LTRIM(STR(orderlst->phase4len))
CASE plenvar="PHASE5LEN"
    ?? LTRIM(STR(orderlst->phase5len))
CASE plenvar="PHASE6LEN"
    ?? LTRIM(STR(orderlst->phase6len))
CASE plenvar="PHASE7LEN"
    ?? LTRIM(STR(orderlst->phase7len))
CASE plenvar="PHASE8LEN"
    ?? LTRIM(STR(orderlst->phase8len))
CASE plenvar="PHASE9LEN"
    ?? LTRIM(STR(orderlst->phase9len))
ENDCASE
? "Parameter Set: "+ordertas->profilenam+"  "+cprofile->profiledes
? "      Profile: "+cprofile->profile
?
IF printdes<>"s" .AND. vttlva<>"o"
    ?
"----------------------------------------------------------------------
--------------------------------"
    ? "|            Equipment Usage Profile in daily hours of operation for all
categories                               |"
```

309

```
   ?  "|             except Wheeled Vehicles which are in daily KMs driven.
                                         |"
   ?  "|
              --Tracked Vehicles-          |"
   ?  "|   AE      AV      CE      GN      HG      MH      OV      SG      SV
   WV     TV_TI   TV_CC   TV_SR        |"
   ?  "|"+TRANSFORM(profiles->ae_hrs,"9999.9")+"   "
   ?? TRANSFORM(profiles->av_hrs,"9999.9")+"   "
   ?? TRANSFORM(profiles->ce_hrs,"9999.9")+"   "
   ?? TRANSFORM(profiles->gn_hrs,"9999.9")+"   "
   ?? TRANSFORM(profiles->hg_hrs,"9999.9")+"   "
   ?? TRANSFORM(profiles->mh_hrs,"9999.9")+"   "
   ?? TRANSFORM(profiles->ov_hrs,"9999.9")+"   "
   ?? TRANSFORM(profiles->sg_hrs,"9999.9")+"   "
   ?? TRANSFORM(profiles->sv_hrs,"9999.9")+"   "
   ?? TRANSFORM(profiles->wv_kms,"9999.9")+"   "
   ?? TRANSFORM(profiles->tv_ti_hrs,"9999.9")+"   "
   ?? TRANSFORM(profiles->tv_cc_hrs,"9999.9")+"   "
   ?? TRANSFORM(profiles->tv_sr_hrs,"9999.9")+"        |"
   ?
  "-------------------------------------------------------------------------
  ------------------------------------"
ENDIF (printdes<>"s" .AND. vttlva<>"o")
*!*********************************************************************
*!
*!       Procedure: DETA_PRO
*!
*!       Called by: PRINTPOL.PRG
*!                : PRINTAMM.PRG
*!                : PRINTOTH.PRG
*!
*!*********************************************************************
PROCEDURE deta_pro
DO CASE
CASE BAR()=1
   detailva="YES"
   DEACTIVATE POPUP
CASE BAR()=2
   detailva="NO"
   vttlva="uto"
   DEACTIVATE POPUP
CASE BAR()=3
   detailva="NO"
   vttlva="to"
   DEACTIVATE POPUP
CASE BAR()=4
   detailva="NO"
   vttlva="o"
   DEACTIVATE POPUP
ENDCASE
CLEAR
* BAR 1 OF deta_pop PROMPT "LIN by LIN Detail"
*!*********************************************************************
*!
*!       Procedure: DETA_BAR1
*!
*!*********************************************************************
PROCEDURE deta_bar1
@ 9,1 CLEAR TO 16,79
@ 9,1
```

```
TEXT
    Consumption data is printed for each equipment LIN, including fuel
    type, fuel tank capacity, gallons consumed, and weight of fuel
    consumed. Unit, Task Organization, and Order summaries follow LIN by
    LIN  detail.  A seperate listing is printed for each Unit if there are
    more than one of a particular Unit.  This can be a very long report,
    and it may be preferred to print it to a file for use with a word
    processor.
ENDTEXT

RETURN
* DEFINE BAR 2 OF deta_pop PROMPT "Summary Report - Unit, Task Org, and Order
Totals"
*!*******************************************************************
*!
*!        Procedure: DETA_BAR2
*!
*!*******************************************************************
PROCEDURE deta_bar2
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
    Consumption data is printed for each fuel type, including gallons
    consumed, fuel tank capacity, and weight of fuel consumed by Unit, Task
    Organization, and Order.
ENDTEXT

RETURN
* DEFINE BAR 3 OF deta_pop PROMPT "Summary Report - Task Org and Order Totals
Only"
*!*******************************************************************
*!
*!        Procedure: DETA_BAR3
*!
*!*******************************************************************
PROCEDURE deta_bar3
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
    Consumption data is printed for each fuel type, including gallons
    consumed, fuel tank capacity, and weight of fuel consumed by Task
    Organization, and Order.
ENDTEXT

RETURN
*DEFINE BAR 4 OF deta_pop PROMPT "Summary Report - Order Totals Only"
*!*******************************************************************
*!
*!        Procedure: DETA_BAR4
*!
*!*******************************************************************
PROCEDURE deta_bar4
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
    Consumption data is printed for each fuel type, including gallons
    consumed, fuel tank capacity, and weight of fuel consumed by the Order.
ENDTEXT

RETURN
```

```
*  BAR 1 OF prin_pop PROMPT "Print Estimate to Printer"
*!***********************************************************************
*!
*!       Procedure: PRIN_BAR1
*!
*!***********************************************************************
PROCEDURE prin_bar1
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
   Make sure printer is on and on-line.
ENDTEXT

RETURN
*  BAR 2 OF prin_pop PROMPT "Print Estimate to Screen"
*!***********************************************************************
*!
*!       Procedure: PRIN_BAR2
*!
*!***********************************************************************
PROCEDURE prin_bar2
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
   Prints Reports to the Screen. These estimates can take a long time
   to view this way.  It may be preferred to print to a file and then
   view the file with a work processor.
ENDTEXT

RETURN
*  BAR 3 OF prin_pop PROMPT "Print Estimate to ASCII File"
*!***********************************************************************
*!
*!       Procedure: PRIN_BAR3
*!
*!***********************************************************************
PROCEDURE prin_bar3
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
   ASCII files can be viewed and edited with most word processors, or
   with EDIT if you are using a version of DOS 5.0 or higher.
ENDTEXT

RETURN
*  BAR 4 OF prin_pop PROMPT "Change Printer Defaults"
*!***********************************************************************
*!
*!       Procedure: PRIN_BAR4
*!
*!***********************************************************************
PROCEDURE prin_bar4
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT
   Change printer page length and left margin.
ENDTEXT

RETURN
*  BAR 5 OF prin_pop PROMPT "Return to Previous Menu"
```

```
*!*************************************************************
*!
*!        Procedure: PRIN_BAR5
*!
*!*************************************************************
PROCEDURE prin_bar5
@ 9,1 CLEAR TO 16,79
@ 9,1
TEXT

ENDTEXT

RETURN

*: EOF: PRINTPOL.PRG
```

```
*:****************************************************************
*:
*:          Program: D:\SURE\PRNSETUP.PRG
*:
*:       Called by: UTIL_PRO               (procedure in SURE.PRG)
*:              : PRIN_PRO               (procedure in SYSREP.PRG)
*:              : SYSR_PRO               (procedure in SYSREP.PRG)
*:
*:   Memory Files: VARIABLE.MEM
*:
*:          Documented 04/16/94 at 11:02              SNAP!  version 5.00
*:****************************************************************
ACTIVATE WINDOW small
@ 0,1 SAY "Left Margin    "
@ ROW(),COL() GET varlmarg PICTURE "99"
IF vprintva<>" "
   @ ROW(), COL() SAY " Printer: "+vprintva
ENDIF (vprintva<>" ")
@ 2,1 SAY "Lines per Page "
@ ROW(),COL() GET varpagel PICTURE "999"
READ
SET MARGIN TO varlmarg
SAVE TO variable ALL LIKE v*
@ 3,1
WAIT
DEACTIVATE WINDOW small
RETURN
*: EOF: PRNSETUP.PRG
```

```
*:********************************************************************
*:
*: Procedure file: D:\SURE\PRNTHELP.PRG
*:
*:  Procs & Fncts: PHEL_PRO
*:               : PRNONEH
*:               : PRNALLH
*:               : ENDPHEL
*:               : BROWHELP
*:
*:         Set by: HELP_PRO              (procedure in SURE.PRG)
*:
*:          Calls: PHEL_PRO              (procedure in PRNTHELP.PRG)
*:
*:       Documented 04/16/94 at 11:02              SNAP!  version 5.00
*:********************************************************************
CLEAR
STORE 0 TO time1var
STORE 2 TO time2var
SET PROCEDURE TO instphlp
SET MEMOWIDTH TO 75-varlmarg
SET MARGIN TO varlmarg
SELECT 39
SET ORDER TO subject
GO TOP
DEFINE POPUP phel_pop FROM 1,5
DEFINE BAR 1 OF phel_pop PROMPT "Print One Help Subject"
DEFINE BAR 2 OF phel_pop PROMPT "Print All Help Subjects"
DEFINE BAR 3 OF phel_pop PROMPT "Return to Main Menu"
ON SELECTION POPUP phel_pop DO phel_pro
ON BAR 1 OF phel_pop DO phel_bar1
ON BAR 2 OF phel_pop DO phel_bar2
ON BAR 3 OF phel_pop DO phel_bar3
DO WHILE PROMPT()<>"Return to Main Menu"
   ACTIVATE POPUP phel_pop
ENDDO (PROMPT()<>"Return to Main Menu")
CLEAR
RETURN
*!********************************************************************
*!
*!       Procedure: PHEL_PRO
*!
*!       Called by: PRNTHELP.PRG
*!
*!           Calls: PRNONEH              (procedure in PRNTHELP.PRG)
*!               : PRNALLH              (procedure in PRNTHELP.PRG)
*!               : ENDPHEL              (procedure in PRNTHELP.PRG)
*!
*!********************************************************************
PROCEDURE phel_pro
DO CASE
CASE BAR()=1
   DO prnoneh
CASE BAR()=2
   DO prnallh
CASE BAR()=3
   DO endphel
ENDCASE
*!********************************************************************
*!
```

315

```
*!          Procedure: PRNONEH
*!
*!          Called by: PHEL_PRO          (procedure in PRNTHELP.PRG)
*!
*!               Calls: BROWHELP          (procedure in PRNTHELP.PRG)
*!                    : PRNTMEMO.PRG
*!
*!**********************************************************************
PROCEDURE prnoneh
SET ORDER TO subject
DO browhelp
_PLENGTH=varpagel
_PLOFFSET=varlmarg
ACTIVATE WINDOW process
_WRAP=.T.
SET PRINT ON
PRINTJOB
   SET DEVICE TO PRINT
   DO prntmemo
ENDPRINTJOB
SET PRINT OFF
_WRAP=.F.
SET DEVICE TO SCREEN
DEACTIVATE WINDOW process
*!**********************************************************************
*!
*!          Procedure: PRNALLH
*!
*!          Called by: PHEL_PRO          (procedure in PRNTHELP.PRG)
*!
*!               Calls: PRNTMEMO.PRG
*!
*!**********************************************************************
PROCEDURE prnallh
SET ORDER TO subject
STORE .T. TO stopvar
GO TOP
@ 22,5 SAY "Press ESC to STOP printing after current subject!"
_PLENGTH=varpagel
_PLOFFSET=varlmarg
_WRAP=.T.
SET PRINT ON
PRINTJOB
   SET ESCAPE OFF
   SET DEVICE TO PRINT
   ACTIVATE WINDOW process
   DO WHILE .NOT. EOF()
      DO prntmemo
      SKIP
      timelvar=VAL(SUBSTR(TIME(),7,2))
      * This looks strange, but for some reason dBase gets confused
      * and does not always leave the do loop. The countvar routine
      * forces an exit to the loop if the time*var routine does not!
      countvar=0
      DO WHILE .T.
         I = INKEY()
         IF I=27
            stopvar=.F.
            EXIT
         ENDIF (I=27)
```

316

```
                time2var=VAL(SUBSTR(TIME(),7,2))
                IF time2var=>time1var+3 .OR. countvar=1000
                    EXIT
                ENDIF (time2var=>time1var+3 .OR. countvar=1000)
                countvar=countvar+1
            ENDDO (.T.)
            IF stopvar = .F.
                EXIT
            ENDIF (stopvar = .F.)
        ENDDO (.NOT. EOF())
    ENDPRINTJOB
    SET PRINT OFF
    SET ESCAPE ON
    _WRAP=.F.
    SET DEVICE TO SCREEN
    GO TOP
    DEACTIVATE WINDOW process
    CLEAR
*!**************************************************************************
*!
*!          Procedure: ENDPHEL
*!
*!          Called by: PHEL_PRO            (procedure in PRNTHELP.PRG)
*!
*!**************************************************************************
PROCEDURE endphel
GO TOP
DEACTIVATE POPUP
RETURN
* Browse Help
*!**************************************************************************
*!
*!          Procedure: BROWHELP
*!
*!          Called by: PRNONEH            (procedure in PRNTHELP.PRG)
*!
*!              Calls: SETBROWS.PRG
*!                   : ENDBROWS.PRG
*!
*!**************************************************************************
PROCEDURE browhelp
ACTIVATE WINDOW browser
DO setbrows
BROWSE NOMENU NOAPPEND NODELETE NOEDIT FIELDS help->subject
DO endbrows
DEACTIVATE WINDOW browser
SET ORDER TO PROGRAM
RETURN

*: EOF: PRNTHELP.PRG
```

```
*:************************************************************************
*:
*:          Program: D:\SURE\PRNTMEMO.PRG
*:
*:        Called by: PRNONEH                (procedure in PRNTHELP.PRG)
*:                 : PRNALLH                (procedure in PRNTHELP.PRG)
*:
*:        Documented 04/16/94 at 11:02                SNAP!  version 5.00
*:************************************************************************
PRIVATE cmemo, numlines,natrow
numlines = MEMLINES(MENU)
IF numlines<>0
   IF PROW()>varpagel-(numlines+5)
      EJECT
   ENDIF (PROW()>varpagel-(numlines+5))
   natline=1
   ? subject
   ?
   DO WHILE natline<=numlines
      ? MLINE(MENU,natline)
      natline=natline+1
      IF PROW()>varpagel-5
         ?
         ? "Page "+TRANSFORM(_PAGENO,"999")
         EJECT
      ENDIF (PROW()>varpagel-5)
   ENDDO (natline<=numlines)
ENDIF (numlines<>0)
numlines = MEMLINES(notes)
IF numlines=0
   RETURN
ENDIF (numlines=0)
natline=2
DO WHILE natline<=numlines
   ? MLINE(notes,natline)
   natline=natline+1
   IF PROW()>varpagel-5
      ?
      ? "Page "+TRANSFORM(_PAGENO,"999")
      EJECT
   ENDIF (PROW()>varpagel-5)
ENDDO (natline<=numlines)
RETURN

*: EOF: PRNTMEMO.PRG
```

```
*:*******************************************************************
*:
*:          Program: D:\SURE\REINDX.PRG
*:
*:          Called by: UTIL_PRO            (procedure in SURE.PRG)
*:
*:             Uses: HELP.DBF
*:                 : CSRCLIST.DBF
*:                 : SRCLIST.DBF
*:                 : SRCLINQT.DBF
*:                 : CSRCLINQ.DBF
*:                 : ECHELONS.DBF
*:                 : LINLIST.DBF
*:                 : AMMORATE.DBF
*:                 : POLRATE.DBF
*:                 : TASKORG.DBF
*:                 : ORDERTAS.DBF
*:                 : MEDICAL.DBF
*:                 : WATER.DBF
*:                 : CLASS_I.DBF
*:                 : PROFILES.DBF
*:                 : CPROFILE.DBF
*:                 : ORDERLST.DBF
*:                 : POLWGT.DBF
*:                 : DODIC.DBF
*:                 : POLEST.DBF
*:                 : AMMOEST.DBF
*:                 : CL_OTHER.DBF
*:
*:          Indexes: PROGRAM             (tag in HELP.MDX)
*:                 : SUBJECT             (tag in HELP.MDX)
*:                 : SRC                 (tag in CSRCLIST.MDX)
*:                 : UNIT_DESIG          (tag in CSRCLIST.MDX)
*:                 : STRENGTH            (tag in CSRCLIST.MDX)
*:                 : UNITNAME            (tag in CSRCLIST.MDX)
*:                 : ECHELON             (tag in CSRCLIST.MDX)
*:                 : LIN                 (tag in SRCLINQT.MDX)
*:                 : UNIQUELI            (tag in SRCLINQT.MDX)
*:                 : DODIC               (tag in AMMORATE.MDX)
*:                 : POL_TYPE            (tag in POLRATE.MDX)
*:                 : TASKORGNAM          (tag in TASKORG.MDX)
*:                 : UNIQUEUN            (tag in TASKORG.MDX)
*:                 : UNIQUETA            (tag in TASKORG.MDX)
*:                 : NUMBER              (tag in ORDERTAS.MDX)
*:                 : TASKPHAS            (tag in ORDERTAS.MDX)
*:                 : PROFILENAM          (tag in ORDERTAS.MDX)
*:                 : TASKPHASE           (tag in ORDERTAS.MDX)
*:                 : UNIQUENU            (tag in ORDERTAS.MDX)
*:                 : PHASE               (tag in ORDERTAS.MDX)
*:                 : UNIQUETN            (tag in ORDERTAS.MDX)
*:                 : PHASETAS            (tag in ORDERTAS.MDX)
*:                 : MED_NAME            (tag in MEDICAL.MDX)
*:                 : WATER_NAME          (tag in WATER.MDX)
*:                 : RATION              (tag in CLASS_I.MDX)
*:                 : PROFILE             (tag in PROFILES.MDX)
*:                 : INTENSITY           (tag in CPROFILE.MDX)
*:                 : PROFILEDES          (tag in CPROFILE.MDX)
*:                 : POSTURE             (tag in CPROFILE.MDX)
*:                 : CDE                 (tag in CPROFILE.MDX)
*:                 : NAME                (tag in ORDERLST.MDX)
```

319

```
*:                   : DESCRIPTIO          (tag in ORDERLST.MDX)
*:                   : PARENT_UNI          (tag in POLEST.MDX)
*:                   : UNITDODI            (tag in AMMOEST.MDX)
*:
*:        MDX files: HELP.MDX
*:                   : CSRCLIST.MDX
*:                   : SRCLIST.MDX
*:                   : SRCLINQT.MDX
*:                   : CSRCLINQ.MDX
*:                   : ECHELONS.MDX
*:                   : LINLIST.MDX
*:                   : AMMORATE.MDX
*:                   : POLRATE.MDX
*:                   : TASKORG.MDX
*:                   : ORDERTAS.MDX
*:                   : MEDICAL.MDX
*:                   : WATER.MDX
*:                   : CLASS_I.MDX
*:                   : PROFILES.MDX
*:                   : CPROFILE.MDX
*:                   : ORDERLST.MDX
*:                   : POLWGT.MDX
*:                   : DODIC.MDX
*:                   : POLEST.MDX
*:                   : AMMOEST.MDX
*:                   : CL_OTHER.MDX
*:
*:        Documented 04/16/94 at 11:02             SNAP!  version 5.00
*:*****************************************************************
* REINDX.PRG Reindexs dbf's
STORE PROGRAM() TO progvar
STORE "NO " TO yesnovar
ACTIVATE WINDOW process
TEXT
 This will recreate all indexes used by SURE. It is not normally
 needed, as SURE's compiled dBase 2.0 system should automatically
 prompt you if it needs to repair an index.  However, if your .MDX
 index files should become deleted and you do not have a current
 backup, this will recreate the .MDX index files.


ENDTEXT

@ ROW(), COL() SAY "Do you wish to reindex? (NO/YES) " GET yesnovar;
   PICTURE "!!!" VALID yesnovar="NO " .OR. yesnovar="YES" ERROR;
   "Must be NO or YES"
READ
IF yesnovar<>"YES"
   DEACTIVATE WINDOW process
   RETURN
ENDIF (yesnovar<>"YES")
SET TALK ON
USE HELP
INDEX ON PROGRAM TAG PROGRAM                       && (PROGRAM)
INDEX ON subject TAG subject                       && (SUBJECT)
USE csrclist
INDEX ON src TAG src
INDEX ON unit_desig TAG unit_desig                 && (UNIT_DESIG)
INDEX ON strength TAG strength                     && (STRENGTH)
INDEX ON unitname TAG unitname                      && (UNITNAME)
```

```
INDEX ON echelon TAG echelon                          && (ECHELON)
USE srclist
INDEX ON src TAG src                                  && (SRC)
INDEX ON unit_desig TAG unit_desig                    && (UNIT_DESIG)
USE srclinqt
INDEX ON src TAG src                                  && (SRC)
INDEX ON lin TAG lin                                  && (LIN)
INDEX ON lin UNIQUE TAG uniqueli                      && (LIN)
USE csrclinq
INDEX ON src TAG src                                  && (SRC)
INDEX ON lin TAG lin                                  && (LIN)
INDEX ON unitname TAG unitname                        && (UNITNAME)
USE echelons
INDEX ON echelon TAG echelon                          && (ECHELON)
USE linlist
INDEX ON lin TAG lin                               && (LIN)
USE ammorate
INDEX ON lin TAG lin                                  && (LIN)
INDEX ON dodic TAG dodic                              && (DODIC)
USE polrate
INDEX ON lin TAG lin                                  && (LIN)
INDEX ON pol_type TAG pol_type                        && (POL_TYPE)
USE taskorg
INDEX ON taskorgnam TAG taskorgnam                    && (TASKORGNAM)
INDEX ON unitname TAG unitname                        && (UNITNAME)
INDEX ON unitname UNIQUE TAG uniqueun                 && (UNITNAME)
INDEX ON taskorgnam UNIQUE TAG uniqueta               && (TASKORGNAM)
INDEX ON src TAG src                                  && (SRC)
USE ordertas
INDEX ON NUMBER TAG NUMBER                            && (NUMBER)
INDEX ON taskorgnam TAG taskorgnam                    && (TASKORGNAM)
INDEX ON taskorgnam+STR(phase) TAG taskphas           && (TASKORGNAM+STR((PHASE))
INDEX ON profilenam TAG profilenam                    && (PROFILENAM)
INDEX ON taskorgnam+STR(phase) TAG taskphase          && (TASKORGNAM+STR((PHASE))
INDEX ON NUMBER UNIQUE TAG uniquenu                   && (NUMBER)
INDEX ON taskorgnam UNIQUE TAG uniqueta               && (TASKORGNAM)
INDEX ON phase TAG phase                              && (PHASE)
INDEX ON taskorgnam+NUMBER TAG uniquetn               && (TASKORGNAM+NUMBER)
INDEX ON STR(phase)+taskorgnam TAG phasetas           && (STR((PHASE)+TASKORGNAM)
USE medical
INDEX ON med_name TAG med_name                        && (MED_NAME)
INDEX ON echelon TAG echelon                          && (ECHELON)
USE water
INDEX ON echelon TAG echelon                          && (ECHELON)
INDEX ON water_name TAG water_name                    && (WATER_NAME)
USE class_i
INDEX ON ration TAG ration                            && (RATION)
USE profiles
INDEX ON profile TAG profile                          && (PROFILE)
USE cprofile
INDEX ON profile TAG profile                          && (PROFILE)
INDEX ON INTENSITY TAG INTENSITY                      && (INTENSITY)
INDEX ON profilenam TAG profilenam                    && (PROFILENAM)
INDEX ON profiledes TAG profiledes                    && (PROFILEDES)
INDEX ON posture TAG posture                          && (POSTURE)
INDEX ON med_name TAG med_name                        && (MED_NAME)
INDEX ON water_name TAG water_name                    && (WATER_NAME)
INDEX ON cde TAG cde                                  && (CDE)
USE orderlst
INDEX ON NUMBER TAG NUMBER                            && (NUMBER)
```

321

```
INDEX ON name TAG name                          && (NAME)
INDEX ON descriptio TAG descriptio          -   && (DESCRIPTIO)
USE polwgt
INDEX ON pol_type TAG pol_type                  && (POL_TYPE)
USE dodic
INDEX ON dodic TAG dodic                         && (DODIC)
USE polest
INDEX ON NUMBER TAG NUMBER                       && (NUMBER)
INDEX ON unitname TAG unitname                   && (UNITNAME)
INDEX ON taskorgnam TAG taskorgnam              && (TASKORGNAM)
INDEX ON lin TAG lin                             && (LIN)
INDEX ON pol_type TAG pol_type                   && (POL_TYPE)
INDEX ON parent_uni TAG parent_uni               && (PARENT_UNI)
USE ammoest
INDEX ON NUMBER TAG NUMBER                       && (NUMBER)
INDEX ON unitname TAG unitname                   && (UNITNAME)
INDEX ON taskorgnam TAG taskorgnam              && (TASKORGNAM)
INDEX ON dodic TAG dodic                         && (DODIC)
INDEX ON lin TAG lin                             && (LIN)
INDEX ON unitname+lin+dodic TAG unitdodi        && (UNITNAME+DODIC)
USE cl_other
INDEX ON unitname TAG unitname                   && (UNITNAME)
INDEX ON taskorgnam TAG taskorgnam              && (TASKORGNAM)
INDEX ON profilenam TAG profilenam              && (PROFILENAM)
SET TALK OFF
? CHR(7)
?
WAIT "All done! Press any key..."
DEACTIVATE WINDOW process
RETURN
*: EOF: REINDX.PRG
```

```
*:*********************************************************************
*:
*: Procedure file: D:\SURE\RESTOREF.PRG
*:
*:   Procs & Fncts: RESTORE
*:                : GETSTATU
*:
*:          Set by: UTIL_PRO              (procedure in SURE.PRG)
*:
*:           Calls: RESTORE              (procedure in RESTOREF.PRG)
*:
*:          Documented 04/16/94 at 11:02              SNAP!  version 5.00
*:*********************************************************************
* restoref.prg
STORE SPACE(8) TO filevar
STORE SPACE(8) TO filevar2
STORE " " TO abortvar
STORE " " TO diskvar
STORE "NO " TO yesnovar
STORE HOME()+SPACE(10) TO homevar
ACTIVATE WINDOW huge
TEXT
 This will copy the SURE files from a previous Backup from your
 floppy drive to the hard drive.  It will OVERWRITE your current files!
 If you do this you will NOT be able to get them back!  Restore is useful
 when:
        1. You have done a lot of entries which have to be reversed, and
           Restoring from your last backup is faster than making changes.

        2. You are copying all of your data from one SURE system to another.
           For example, you might use one system in the office, another in
           the field.

 It is HIGHLY recommended that you do a seperate backup of your current
 SURE files first!


ENDTEXT
@ ROW(), COL() SAY "Do you want to continue to Restore? (NO/YES) "
@ ROW(), COL() GET yesnovar PICTURE "!!!" VALID yesnovar="NO" .OR.
yesnovar="YES";
     ERROR "Must be NO or YES"
READ
IF yesnovar<>"YES"
     DEACTIVATE WINDOW huge
     RETURN
ENDIF (yesnovar<>"YES")
CLEAR
TEXT
 Sure thinks that your SURE system is in the following drive and directory.
 If you are sure that they are wrong, change them at this time, or Return
 to continue with the Restore process.


ENDTEXT

@ ROW(), COL() GET homevar
READ
homevar=RTRIM(homevar)
ACTIVATE WINDOW small
```

323

```
@ 1,1 SAY "Enter disk drive to Restore from or"
@ 2,1 SAY "Return to Quit: "
@ ROW(), COL() GET diskvar
READ
IF diskvar=" "
   DEACTIVATE WINDOW small
   RETURN
ENDIF (diskvar=" ")
DEACTIVATE WINDOW small
CLEAR
@ 1,2 SAY "Please wait, restoring..."
diskvar=diskvar+":\"
? homevar
SET MESSAGE TO "Restore from Floppy Disk"
filevar=diskvar+"CSRCLIST.DBF"
filevar2=homevar+"CSRCLIST.DBF"
DO RESTORE
filevar=diskvar+"CSRCLINQ.DBF"
filevar2=homevar+"CSRCLINQ.DBF"
DO RESTORE
filevar=diskvar+"TASKORG.DBF"
filevar2=homevar+"TASKORG.DBF"
DO RESTORE
filevar=diskvar+"ORDERTAS.DBF"
filevar2=homevar+"ORDERTAS.DBF"
DO RESTORE
filevar=diskvar+"PROFILES.DBF"
filevar2=homevar+"PROFILES.DBF"
DO RESTORE
filevar=diskvar+"INTENSTY.DBF"
filevar2=homevar+"INTENSTY.DBF"
DO RESTORE
filevar=diskvar+"CPROFILE.DBF"
filevar2=homevar+"CPROFILE.DBF"
DO RESTORE
filevar=diskvar+"ORDERLST.DBF"
filevar2=homevar+"ORDERLST.DBF"
DO RESTORE
filevar=diskvar+"CSRCLIST.MDX"
filevar2=homevar+"CSRCLIST.MDX"
DO RESTORE
filevar=diskvar+"CSRCLINQ.MDX"
filevar2=homevar+"CSRCLINQ.MDX"
DO RESTORE
filevar=diskvar+"TASKORG.MDX"
filevar2=homevar+"TASKORG.MDX"
DO RESTORE
filevar=diskvar+"ORDERTAS.MDX"
filevar2=homevar+"ORDERTAS.MDX"
DO RESTORE
filevar=diskvar+"PROFILES.MDX"
filevar2=homevar+"PROFILES.MDX"
DO RESTORE
filevar=diskvar+"INTENSTY.MDX"
filevar2=homevar+"INTENSTY.MDX"
DO RESTORE
filevar=diskvar+"ORDERLST.MDX"
filevar2=homevar+"ORDERLST.MDX"
DO RESTORE
DEACTIVATE WINDOW huge
```

```
RETURN
*!********************************************************************
*!
*!      Procedure: RESTORE
*!
*!      Called by: RESTOREF.PRG
*!
*!          Calls: GETSTATU              (procedure in RESTOREF.PRG)
*!
*!           Uses: &FILEVAR.DBF
*!              : &FILEVAR2.DBF
*!
*!********************************************************************
PROCEDURE RESTORE
DO WHILE .T.
   @ 3,2 SAY "                    "
   @ 3,2 SAY filevar
   IF FILE(filevar)
       COPY FILE &filevar TO &filevar2
       EXIT
   ELSE
       DO getstatu
       DO CASE
       CASE yesnovar="T"
          LOOP
       CASE yesnovar="C"
          EXIT
       CASE yesnovar="Q"
          *DEACTIVATE WINDOW process
          DEACTIVATE WINDOW huge
          SET PROCEDURE TO INSTRUCT
          RETURN TO sure
       ENDCASE
       DEACTIVATE WINDOW process
   ENDIF (FILE(filevar))
ENDDO (.T.)
*!********************************************************************
*!
*!      Procedure: GETSTATU
*!
*!      Called by: RESTORE              (procedure in RESTOREF.PRG)
*!
*!********************************************************************
PROCEDURE getstatu
yesnovar=" "
? CHR(7)
CLEAR
*ACTIVATE WINDOW process
@ 1,1 SAY filevar
TEXT
 NOT FOUND! If you have it on another disk, change disks at this time
 and enter T to Try Again.  If you want to skip this file, enter C to
 Continue.  If you want to stop the Restore process, enter Q to Quit.

ENDTEXT
@ ROW(), COL() SAY "T/C/Q ? " GET yesnovar PICTURE "!" VALID yesnovar="T" .OR.;
yesnovar="C" .OR. yesnovar="Q" ERROR "Must be T, C or Q"
READ
CLEAR
```

```
*: EOF: RESTOREF.PRG
```

```
*:******************************************************************
*:
*:        Program: D:\SURE\SETBROW2.PRG
*:
*:      Called by: VIEWTASK              (procedure in EDITTASK.PRG)
*:               : VIEWTAS2              (procedure in EDITTASK.PRG)
*:               : PHASEBYS              (procedure in EDITORDE.PRG)
*:
*:        Documented 04/16/94 at 11:02                SNAP!  version 5.00
*:******************************************************************
ON KEY LABEL HOME KEYBOARD CHR(31)
ON KEY LABEL end KEYBOARD CHR(30)
SET MESSAGE TO " or  to move, Home for first record, End for last, Ctrl-W when
Done"
*: EOF: SETBROW2.PRG
```

```
*:***************************************************************
*:
*:         Program:  D:\SURE\SETBROWS.PRG
*:
*:         Called by: HELPNDX.PRG
*:                  : EDITHELP.PRG
*:                  : COPYSRC          (procedure in CREATESR.PRG)
*:                  : COPYCSRC         (procedure in CREATESR.PRG)
*:                  : VIEWUNIT         (procedure in CREATESR.PRG)
*:                  : ADDDATA          (procedure in CREATESR.PRG)
*:                  : SELECTUNI        (procedure in EDITUNIT.PRG)
*:                  : SELECTUN2        (procedure in EDITUNIT.PRG)
*:                  : CHANGELI         (procedure in EDITUNIT.PRG)
*:                  : ADDLINCA         (procedure in EDITUNIT.PRG)
*:                  : DELLIN           (procedure in EDITUNIT.PRG)
*:                  : CHANGEEH         (procedure in EDITUNIT.PRG)
*:                  : FINDTA           (procedure in CREATETC.PRG)
*:                  : NEWPROFI         (procedure in CREATEPR.PRG)
*:                  : VIEWPROF         (procedure in CREATEPR.PRG)
*:                  : SELEPROF         (procedure in EDITPROF.PRG)
*:                  : CHANGEAP         (procedure in EDITPROF.PRG)
*:                  : CHANGEAC         (procedure in EDITPROF.PRG)
*:                  : CHANGEI          (procedure in EDITPROF.PRG)
*:                  : CHANGECI         (procedure in EDITPROF.PRG)
*:                  : CHANGEH2         (procedure in EDITPROF.PRG)
*:                  : CHANGEME         (procedure in EDITPROF.PRG)
*:                  : SELECTPRO        (procedure in DELPROF.PRG)
*:                  : NEWCUSPR         (procedure in CRECUSPR.PRG)
*:                  : VIEWCUSP         (procedure in CRECUSPR.PRG)
*:                  : EDITCUSP         (procedure in CRECUSPR.PRG)
*:                  : DELCUSPR         (procedure in CRECUSPR.PRG)
*:                  : VIEWORDE         (procedure in CREATEOR.PRG)
*:                  : 1PROONE          (procedure in CREATEOR.PRG)
*:                  : VIEWTOP          (procedure in CREATEOR.PRG)
*:                  : FINDPRO          (procedure in CREATEOR.PRG)
*:                  : GETORDER         (procedure in EDITORDE.PRG)
*:                  : 1STRONE          (procedure in EDITORDE.PRG)
*:                  : VIEWSTR          (procedure in EDITORDE.PRG)
*:                  : GETSRC           (procedure in SYSREP.PRG)
*:                  : BROWHELP         (procedure in PRNTHELP.PRG)
*:
*:         Documented 04/16/94 at 11:02          SNAP!  version 5.00
*:***************************************************************
ON KEY LABEL HOME KEYBOARD CHR(31)
ON KEY LABEL end KEYBOARD CHR(30)
ON KEY LABEL ctrl-m KEYBOARD CHR(23)
SET MESSAGE TO " or  to move, Home for first record, End for last, Enter to
Select"
*: EOF: SETBROWS.PRG
```

```
*:*****************************************************************
*:'
*:         Program: D:\SURE\SETCON.PRG
*:
*:       Called by: SURE.PRG
*:
*:         Documented 04/16/94 at 11:01            SNAP!  version 5.00
*:*****************************************************************
* SETCON.prg
* SURE 'SET' conditions program.
* CALLED BY: SURE.prg
* CALLS:
* AUTHOR: John M. Friedson
* COPYWRITE: U.S. Army
* Date 15 December 1993
* Set program conditions
SET CENTURY ON
SET BELL OFF
SET EXCLUSIVE ON
SET SCOREBOARD OFF
SET SAFETY OFF
SET TALK OFF
RETURN
*: EOF: SETCON.PRG
```

```
*:****************************************************************
*:
*:          Program: D:\SURE\STARTHLP.PRG
*:
*:        Called by: SURE.PRG
*:                  : SRC_PRO              (procedure in SURE.PRG)
*:                  : TASK_PRO             (procedure in SURE.PRG)
*:                  : PROF_PRO             (procedure in SURE.PRG)
*:                  : ORDE_PRO             (procedure in SURE.PRG)
*:                  : REPO_PRO             (procedure in SURE.PRG)
*:                  : UTIL_PRO             (procedure in SURE.PRG)
*:                  : HELP_PRO             (procedure in SURE.PRG)
*:                  : DELORDE.PRG
*:                  : STARTCSR             (procedure in CREATESR.PRG)
*:                  : STARTUNI             (procedure in EDITUNIT.PRG)
*:                  : STARTED              (procedure in DELUNIT.PRG)
*:                  : STARTASK             (procedure in CREATETA.PRG)
*:                  : STARTPRO             (procedure in CREATEPR.PRG)
*:                  : STARTEPR             (procedure in EDITPROF.PRG)
*:                  : STARTOR              (procedure in CREATEOR.PRG)
*:                  : CALCPOL.PRG
*:                  : CALCAMMO.PRG
*:                  : CALCOTHE.PRG
*:                  : STARTSRP             (procedure in SYSREP.PRG)
*:                  : STARTPLR             (procedure in PRINTPOL.PRG)
*:
*:            Calls: HELP.PRG
*:
*:             Uses: HELP.DBF
*:
*:        MDX files: HELP.MDX
*:
*:        Documented 04/16/94 at 11:01              SNAP!  version 5.00
*:****************************************************************
SELECT 39
USE HELP ORDER PROGRAM
ON KEY LABEL f1 DO HELP
RETURN
*: EOF: STARTHLP.PRG
```

```
*:********************************************************************
*:
*:  Procedure file: D:\SURE\SURE.PRG
*:
*:     Procs & Fncts: DEF_MENS
*:                   : SRC_PRO
*:                   : TASK_PRO
*:                   : PROF_PRO
*:                   : ORDE_PRO
*:                   : REPO_PRO
*:                   : UTIL_PRO
*:                   : QUIT_PRO
*:                   : HELP_PRO
*:
*:            Calls: HELP.PRG
*:                   : SETCON.PRG
*:                   : STARTHLP.PRG
*:                   : MEMVARS.PRG
*:                   : DEF_MENS              (procedure in SURE.PRG)
*:
*:    Memory Files: VARIABLE.MEM
*:
*:        Documented 04/16/94 at 11:01              SNAP!  version 5.00
*:********************************************************************
* SURE.prg MAIN SURE SYSTEM PROGRAM
* CALLED BY:
* CALLS: setcon.prg memvars.prg fileopen.prg
* AUTHOR: John M. Friedson
* COPYWRITE: U.S. Army
* Date 15 December 1993
* Make sure insert key is turned on
DO setcon
* Public Variables
PUBLIC numberva, varlmarg, varpagel, progvar, progvar2, printvar
PUBLIC filevar
STORE SPACE(60) TO filevar
RESTORE FROM variable ADDITIVE
STORE " " TO numberva
STORE SPACE(10) TO printvar
IF ISCOLOR()
    SET COLOR TO W+/B,B/W+
ENDIF (ISCOLOR())
CLEAR
IF vprintva="Laser"
    _PEJECT="AFTER"
ELSE
    _PEJECT="NONE"
ENDIF (vprintva="Laser")
SET PROCEDURE TO instruct.prg
DO starthlp
STORE PROGRAM() TO progvar2
STORE PROGRAM() TO progvar
LOAD isinsert
tempvar = "x"
CALL isinsert WITH tempvar
*IF tempvar<>"T"
KEYBOARD CHR(22)
*ENDIF
*DEFINE universal items
DEFINE WINDOW browser FROM 1,1 TO 15,77
```

331

```
DEFINE WINDOW editor FROM 5,2 TO 22,77
DEFINE WINDOW small FROM 12,20 TO 18,60
DEFINE WINDOW medium FROM 1,1 TO 16,77
DEFINE WINDOW huge FROM 1,1 TO 23,77
DEFINE WINDOW process FROM 2,2 TO 14,77
* Initialize memory variables
DO memvars
* Display Credits
@ 19,1 TO 19,77
@ 20,12 SAY "SUPPLY USAGE REQUIREMENTS ESTIMATOR (SURE) VERSION 2.0"
@ 21,12 SAY " This is WORK IN PROGRESS... do NOT use this version!"
@ 22,0 SAY "          Developed by CASCOM and the School for Advanced Military
Studies"
@ 23,0 SAY "F1=Help                  Major John M Friedson, QM Programmer"
*@ 23,60 SAY "Memory: "
*@ ROW(),COL() SAY LTRIM(STR(MEMORY()))
* Open database files
* DO openfile
* Define Menu Items
DO def_mens
* Assign POPUPs to the Pads
ON PAD src OF main ACTIVATE POPUP src_pop
ON PAD task OF main ACTIVATE POPUP task_pop
ON PAD profiles OF main ACTIVATE POPUP prof_pop
ON PAD orders OF main ACTIVATE POPUP orde_pop
ON PAD reports OF main ACTIVATE POPUP repo_pop
ON PAD utility OF main ACTIVATE POPUP util_pop
ON PAD QUIT OF main ACTIVATE POPUP quit_pop
ON PAD HELP OF main ACTIVATE POPUP help_pop
DO WHILE PROMPT()<>"QUIT"
   ACTIVATE MENU main PAD src
ENDDO (PROMPT()<>"QUIT")
CLEAR ALL
SET TALK ON
RETURN
* Create Main menu and define its Pads
*!*****************************************************************
*!
*!        Procedure: DEF_MENS
*!
*!        Called by: SURE.PRG
*!
*!            Calls: SRC_PRO         (procedure in SURE.PRG)
*!                 : TASK_PRO        (procedure in SURE.PRG)
*!                 : PROF_PRO        (procedure in SURE.PRG)
*!                 : ORDE_PRO        (procedure in SURE.PRG)
*!                 : REPO_PRO        (procedure in SURE.PRG)
*!                 : UTIL_PRO        (procedure in SURE.PRG)
*!                 : HELP_PRO        (procedure in SURE.PRG)
*!                 : QUIT_PRO        (procedure in SURE.PRG)
*!
*!*****************************************************************
PROCEDURE def_mens
DEFINE MENU main
DEFINE PAD src OF main PROMPT "Units"
DEFINE PAD task OF main PROMPT "Task Organizations"
DEFINE PAD profiles OF main PROMPT "Parameter Sets"
DEFINE PAD orders OF main PROMPT "Orders"
DEFINE PAD reports OF main PROMPT "Reports"
DEFINE PAD utility OF main PROMPT "Utilities"
```

332

```
DEFINE PAD HELP OF main PROMPT "Help"
DEFINE PAD QUIT OF main PROMPT "QUIT"
* Define the SRC pad's popup and bars
DEFINE POPUP src_pop FROM 1,0 MESSAGE "Unit"
DEFINE BAR 1 OF src_pop PROMPT "Create Unit from SRC"
DEFINE BAR 2 OF src_pop PROMPT "Edit Unit"
DEFINE BAR 3 OF src_pop PROMPT "Create Custom Unit from Equipment LINs"
DEFINE BAR 5 OF src_pop PROMPT "Delete Unit"
ON SELECTION POPUP src_pop DO src_pro
* ON the SRC pad's  bars
ON BAR 1 OF src_pop DO surecsr1
ON BAR 2 OF src_pop DO surecsr2
ON BAR 3 OF src_pop DO surecsr3
ON BAR 5 OF src_pop DO surecsr5
* Define the Task pad's popup and bars
DEFINE POPUP task_pop FROM 1,6 MESSAGE "Task Organization"
DEFINE BAR 1 OF task_pop PROMPT "Create Task Organization from Units"
DEFINE BAR 2 OF task_pop PROMPT "Create Task Organization by Copying a Previous
One"
DEFINE BAR 3 OF task_pop PROMPT "Edit Task Organization"
DEFINE BAR 5 OF task_pop PROMPT "Delete Task Organization"
ON SELECTION POPUP task_pop DO task_pro
* ON the Task pad's  bars
ON BAR 1 OF task_pop DO suretas1
ON BAR 2 OF task_pop DO suretas2
ON BAR 3 OF task_pop DO suretas3
ON BAR 5 OF task_pop DO suretas5
* Define the Parameter Sets pad's popup and bars
DEFINE POPUP prof_pop FROM 1,25 MESSAGE "Mission Parameter Sets"
DEFINE BAR 1 OF prof_pop PROMPT "Create Mission Parameter Set"
DEFINE BAR 2 OF prof_pop PROMPT "Edit Mission Parameter Set"
DEFINE BAR 4 OF prof_pop PROMPT "Delete Mission Parameter Set"
DEFINE BAR 6 OF prof_pop PROMPT "Custom Area Profiles"
ON SELECTION POPUP prof_pop DO prof_pro
* ON the Profiles pad's  bars
ON BAR 1 OF prof_pop DO surepro1
ON BAR 2 OF prof_pop DO surepro2
ON BAR 4 OF prof_pop DO surepro4
ON BAR 6 OF prof_pop DO surepro6
* Define the Orders pad's popup and bars
DEFINE POPUP orde_pop FROM 1,40 MESSAGE "Orders"
DEFINE BAR 1 OF orde_pop PROMPT "Create Order"
DEFINE BAR 2 OF orde_pop PROMPT "Edit Order"
DEFINE BAR 4 OF orde_pop PROMPT "Delete Order"
ON SELECTION POPUP orde_pop DO orde_pro
* ON the Orders pad's  bars
ON BAR 1 OF orde_pop DO sureord1
ON BAR 2 OF orde_pop DO sureord2
ON BAR 4 OF orde_pop DO sureord4
* Define the Reports pad's popup and bars
DEFINE POPUP repo_pop FROM 1,47 MESSAGE "Reports"
DEFINE BAR 1 OF repo_pop PROMPT "Orders-based Reports"
*    DEFINE BAR 2 OF repo_pop PROMPT "SRC-based Surelite System"
*    DEFINE BAR 3 OF repo_pop PROMPT "Population-based Reports"
DEFINE BAR 2 OF repo_pop PROMPT "System Reports"
ON SELECTION POPUP repo_pop DO repo_pro
* ON the Reports pad's  bars
ON BAR 1 OF repo_pop DO surerep1
*    ON BAR 2 OF repo_pop DO surerep2
*    ON BAR 3 OF repo_pop DO surerep3
```

333

```
ON BAR 2 OF repo_pop DO surerep2
* Define the Utilities pad's popup and bars
DEFINE POPUP util_pop FROM 1,55 MESSAGE "Utilities"
DEFINE BAR 1 OF util_pop PROMPT "Reindex Files"
DEFINE BAR 2 OF util_pop PROMPT "Backup Files"
DEFINE BAR 3 OF util_pop PROMPT "Restore Files"
DEFINE BAR 4 OF util_pop PROMPT "Printer Setup"
DEFINE BAR 5 OF util_pop PROMPT "Compressed Print ON"
DEFINE BAR 6 OF util_pop PROMPT "Compressed Print OFF"
ON SELECTION POPUP util_pop DO util_pro
* ON the Utilities pad's bars
ON BAR 1 OF util_pop DO sureuti1
ON BAR 2 OF util_pop DO sureuti2
ON BAR 3 OF util_pop DO sureuti3
ON BAR 4 OF util_pop DO sureuti4
ON BAR 5 OF util_pop DO sureuti5
ON BAR 6 OF util_pop DO sureuti6
* Define the HELP pad's popup and bars
DEFINE POPUP help_pop FROM 1,54 MESSAGE "HELP"
DEFINE BAR 1 OF help_pop PROMPT "Browse Help"
DEFINE BAR 2 OF help_pop PROMPT "Print Instructions"
DEFINE BAR 3 OF help_pop PROMPT "Edit Help Database"
ON SELECTION POPUP help_pop DO help_pro
* ON the HELP pad's bars
ON BAR 1 OF help_pop DO surehel1
ON BAR 2 OF help_pop DO surehel2
* Define the Quit pad's popup and bars
DEFINE POPUP quit_pop FROM 1,61 MESSAGE "QUIT"
DEFINE BAR 1 OF quit_pop PROMPT "Do NOT Quit!"
DEFINE BAR 2 OF quit_pop PROMPT "QUIT"
ON SELECTION POPUP quit_pop DO quit_pro
* ON the Quit pad's bars
ON BAR 1 OF quit_pop DO surequi1
ON BAR 2 OF quit_pop DO surequi2
RETURN
*!********************************************************************
*!
*!        Procedure: SRC_PRO
*!
*!        Called by: DEF_MENS              (procedure in SURE.PRG)
*!
*!            Calls: CREATESR.PRG
*!                 : STARTHLP.PRG
*!                 : EDITUNIT.PRG
*!                 : CUSTUNIT.PRG
*!                 : DELUNIT.PRG
*!
*!********************************************************************
PROCEDURE src_pro
DO CASE
CASE BAR() = 1
   SAVE SCREEN TO screenva
   CLOSE PROCEDURE
   progvar2=progvar
   DO createsr
   CLOSE PROCEDURE
   SET PROCEDURE TO INSTRUCT
   RESTORE SCREEN FROM screenva
   DO starthlp
   progvar=progvar2
```

334

```
   CASE BAR() = 2
      SAVE SCREEN TO screenva
      CLOSE PROCEDURE
      DO editunit
      CLOSE PROCEDURE
      SET PROCEDURE TO INSTRUCT
      RESTORE SCREEN FROM screenva
      DO starthlp
      progvar=progvar2
   CASE BAR() = 3
      SAVE SCREEN TO screenva
      CLOSE PROCEDURE
      DO custunit
      CLOSE PROCEDURE
      SET PROCEDURE TO INSTRUCT
      RESTORE SCREEN FROM screenva
      DO starthlp
      progvar=progvar2
   CASE BAR() = 5
      SAVE SCREEN TO screenva
      CLOSE PROCEDURE
      DO delunit
      CLOSE PROCEDURE
      SET PROCEDURE TO INSTRUCT
      RESTORE SCREEN FROM screenva
      DO starthlp
      progvar=progvar2
ENDCASE
*!*********************************************************************
*!
*!        Procedure: TASK_PRO
*!
*!        Called by: DEF_MENS            (procedure in SURE.PRG)
*!
*!            Calls: CREATETA.PRG
*!                 : STARTHLP.PRG
*!                 : CREATETC.PRG
*!                 : EDITTASK.PRG
*!                 : DELTASK.PRG
*!
*!*********************************************************************
PROCEDURE task_pro
DO CASE
CASE BAR() = 1
   SAVE SCREEN TO screenva
   CLOSE PROCEDURE
   DO createta
   CLOSE PROCEDURE
   SET PROCEDURE TO INSTRUCT
   RESTORE SCREEN FROM screenva
   DO starthlp
   progvar=progvar2
CASE BAR() = 2
   SAVE SCREEN TO screenva
   CLOSE PROCEDURE
   DO createtc
   CLOSE PROCEDURE
   SET PROCEDURE TO INSTRUCT
   RESTORE SCREEN FROM screenva
   DO starthlp
```

335

```
      progvar=progvar2
CASE BAR() = 3
   SAVE SCREEN TO screenva
   CLOSE PROCEDURE
   DO edittask
   CLOSE PROCEDURE
   SET PROCEDURE TO INSTRUCT
   RESTORE SCREEN FROM screenva
   DO starthlp
   progvar=progvar2
CASE BAR() = 5
   SAVE SCREEN TO screenva
   CLOSE PROCEDURE
   DO deltask
   CLOSE PROCEDURE
   SET PROCEDURE TO INSTRUCT
   RESTORE SCREEN FROM screenva
   DO starthlp
   progvar=progvar2
ENDCASE
RETURN
*!**************************************************************************
*!
*!        Procedure: PROF_PRO
*!
*!        Called by: DEF_MENS            (procedure in SURE.PRG)
*!
*!            Calls: CREATEPR.PRG
*!                 : STARTHLP.PRG
*!                 : EDITPROF.PRG
*!                 : DELPROF.PRG
*!                 : CRECUSPR.PRG
*!
*!**************************************************************************
PROCEDURE prof_pro
DO CASE
CASE BAR() = 1
   SAVE SCREEN TO screenva
   CLOSE PROCEDURE
   DO createpr
   CLOSE PROCEDURE
   SET PROCEDURE TO INSTRUCT
   RESTORE SCREEN FROM screenva
   DO starthlp
   progvar=progvar2
CASE BAR() = 2
   SAVE SCREEN TO screenva
   CLOSE PROCEDURE
   DO editprof
   CLOSE PROCEDURE
   SET PROCEDURE TO INSTRUCT
   RESTORE SCREEN FROM screenva
   DO starthlp
   progvar=progvar2
CASE BAR() = 4
   SAVE SCREEN TO screenva
   CLOSE PROCEDURE
   DO delprof
   CLOSE PROCEDURE
   SET PROCEDURE TO INSTRUCT
```

336

```
      RESTORE SCREEN FROM screenva
      DO starthlp
      progvar=progvar2
CASE BAR() = 6
      SAVE SCREEN TO screenva
      CLOSE PROCEDURE
      DO crecuspr
      CLOSE PROCEDURE
      SET PROCEDURE TO INSTRUCT
      RESTORE SCREEN FROM screenva
      DO starthlp
      progvar=progvar2
ENDCASE
RETURN
*!*****************************************************************
*!
*!      Procedure: ORDE_PRO
*!
*!      Called by: DEF_MENS              (procedure in SURE.PRG)
*!
*!          Calls: CREATEOR.PRG
*!               : STARTHLP.PRG
*!               : EDITORDE.PRG
*!               : DELORDE.PRG
*!
*!*****************************************************************
PROCEDURE orde_pro
DO CASE
CASE BAR() = 1
      SAVE SCREEN TO screenva
      CLEAR
      CLOSE PROCEDURE
      DO createor
      CLOSE PROCEDURE
      SET PROCEDURE TO INSTRUCT
      RESTORE SCREEN FROM screenva
      DO starthlp
      progvar=progvar2
CASE BAR() = 2
      SAVE SCREEN TO screenva
      CLOSE PROCEDURE
      DO editorde
      CLOSE PROCEDURE
      SET PROCEDURE TO INSTRUCT
      RESTORE SCREEN FROM screenva
      DO starthlp
      progvar=progvar2
CASE BAR() = 4
      SAVE SCREEN TO screenva
      CLOSE PROCEDURE
      DO delorde
      CLOSE PROCEDURE
      SET PROCEDURE TO INSTRUCT
      RESTORE SCREEN FROM screenva
      DO starthlp
      progvar=progvar2
ENDCASE
RETURN
*!*****************************************************************
*!
```

337

```
*!          Procedure: REPO_PRO
*!
*!          Called by: DEF_MENS          (procedure in SURE.PRG)
*!
*!               Calls: ORDEREP1.PRG
*!                    : STARTHLP.PRG
*!                    : SYSREP.PRG
*!
*!*******************************************************************
PROCEDURE repo_pro
DO CASE
CASE BAR() = 1
   SAVE SCREEN TO screenva
   CLOSE PROCEDURE
   DO orderep1
   CLOSE PROCEDURE
   SET PROCEDURE TO INSTRUCT
   RESTORE SCREEN FROM screenva
   DO starthlp
   progvar=progvar2
   *    CASE BAR() = 2
   *        SAVE SCREEN TO screenva
   *        CLOSE PROCEDURE
   *        DO surelit2
   *        CLOSE PROCEDURE
   *        SET PROCEDURE TO INSTRUCT
   *        RESTORE SCREEN FROM screenva
   *            DO starthlp
   *        progvar=progvar2
   *    CASE BAR() = 3
   *        SAVE SCREEN TQ screenva
   *        CLOSE PROCEDURE
   *        DO poprep
   *        CLOSE PROCEDURE
   *        SET PROCEDURE TO INSTRUCT
   *        RESTORE SCREEN FROM screenva
CASE BAR() = 2
   SAVE SCREEN TO screenva
   CLOSE PROCEDURE
   DO sysrep
   CLOSE PROCEDURE
   SET PROCEDURE TO INSTRUCT
   RESTORE SCREEN FROM screenva
ENDCASE
DO starthlp
progvar=progvar2
RETURN
*!*******************************************************************
*!
*!          Procedure: UTIL_PRO
*!
*!          Called by: DEF_MENS          (procedure in SURE.PRG)
*!
*!               Calls: REINDX.PRG
*!                    : STARTHLP.PRG
*!                    : BACKDBF.PRG
*!                    : RESTOREF.PRG
*!                    : PRNSETUP.PRG
*!                    : COMPRESS.PRG
*!                    : UNCOMPRE.PRG
```

338

```
*!
*!*******************************************************************
PROCEDURE util_pro
DO CASE
CASE BAR() = 1
    SAVE SCREEN TO screenva
    CLOSE PROCEDURE
    DO reindx
    CLOSE PROCEDURE
    SET PROCEDURE TO INSTRUCT
    RESTORE SCREEN FROM screenva
    DO starthlp
    progvar=progvar2
CASE BAR() = 2
    SAVE SCREEN TO screenva
    CLOSE PROCEDURE
    DO backdbf
    CLOSE PROCEDURE
    SET PROCEDURE TO INSTRUCT
    RESTORE SCREEN FROM screenva
    DO starthlp
    progvar=progvar2
CASE BAR() = 3
    SAVE SCREEN TO screenva
    CLOSE PROCEDURE
    DO restoref
    CLOSE PROCEDURE
    SET PROCEDURE TO INSTRUCT
    RESTORE SCREEN FROM screenva
    DO starthlp
    progvar=progvar2
CASE BAR() = 4
    DO prnsetup
    DO starthlp
    progvar=progvar2
CASE BAR() = 5
    SAVE SCREEN TO screenva
    CLOSE PROCEDURE
    DO COMPRESS
    CLOSE PROCEDURE
    SET PROCEDURE TO INSTRUCT
    RESTORE SCREEN FROM screenva
    DO starthlp
    progvar=progvar2
CASE BAR() = 6
    SAVE SCREEN TO screenva
    CLOSE PROCEDURE
    DO uncompre
    CLOSE PROCEDURE
    SET PROCEDURE TO INSTRUCT
    RESTORE SCREEN FROM screenva
    DO starthlp
    progvar=progvar2
ENDCASE
RETURN
*!*******************************************************************
*!
*!        Procedure: QUIT_PRO
*!
*!        Called by: DEF_MENS            (procedure in SURE.PRG)
```

339

```
*!
*!           Calls: CLOSEALL.PRG
*!
*!**********************************************************************
PROCEDURE quit_pro
DO CASE
CASE BAR() = 1
   SAVE SCREEN TO screenva
   RESTORE SCREEN FROM screenva
CASE BAR() = 2
   SAVE SCREEN TO screenva
   DO closeall
   SET PROCEDURE TO INSTRUCT
   RESTORE SCREEN FROM screenva
ENDCASE
RETURN
*!**********************************************************************
*!
*!        Procedure: HELP_PRO
*!
*!        Called by: DEF_MENS              (procedure in SURE.PRG)
*!
*!           Calls: HELPNDX.PRG
*!                : STARTHLP.PRG
*!                : PRNTHELP.PRG
*!                : EDITHELP.PRG
*!**********************************************************************
PROCEDURE help_pro
DO CASE
CASE BAR() = 1
   SAVE SCREEN TO screenva
   CLOSE PROCEDURE
   DO helpndx
   CLOSE PROCEDURE
   SET PROCEDURE TO INSTRUCT
   RESTORE SCREEN FROM screenva
   DO starthlp
   progvar=progvar2
CASE BAR() = 2
   SAVE SCREEN TO screenva
   CLOSE PROCEDURE
   DO prnthelp
   CLOSE PROCEDURE
   SET PROCEDURE TO INSTRUCT
   RESTORE SCREEN FROM screenva
   DO starthlp
   progvar=progvar2
CASE BAR() = 3
   SAVE SCREEN TO screenva
   CLOSE PROCEDURE
   DO edithelp
   CLOSE PROCEDURE
   SET PROCEDURE TO INSTRUCT
   RESTORE SCREEN FROM screenva
   DO starthlp
   progvar=progvar2
ENDCASE
RETURN

*: EOF: SURE.PRG
```

340

```
*:********************************************************************.***
*:
*: Procedure file: D:\SURE\SYSREP.PRG
*:
*:   Procs & Fncts: STARTSRP
*:                : PRIN_PRO
*:                : SYSR_PRO
*:                : SRCRPT
*:                : SRCLST
*:                : SRCLINR
*:                : LINLST
*:                : UNITRPT
*:                : UNITLINR
*:                : PROFILER
*:                : TASKORGR
*:                : TASKUNIT
*:                : ORDERRPT
*:                : ORDERTAS
*:                : ORDTASUN
*:                : SYSREND
*:                : GETSRC
*:                : DISPSRC
*:                : LISTSRC
*:                : LISTPROF
*:                : UNITLST
*:                : UNITLIN
*:                : PROFLST
*:                : TASKLST
*:                : TASKUNI
*:                : ORDERTP
*:                : ORDERLST
*:                : ORDERTUP
*:
*:          Set by: REPO_PRO              (procedure in SURE.PRG)
*:
*:          Calls: STARTSRP              (procedure in SYSREP.PRG)
*:               : PRIN_PRO              (procedure in SYSREP.PRG)
*:               : SYSR_PRO              (procedure in SYSREP.PRG)
*:
*:       Documented 04/16/94 at 11:02              SNAP!  version 5.00
*:********************************************************************
* SRCrep.prg SRC-based quick logistics estimations
STORE PROGRAM() TO progvar
* PRIVATE Variables
** Temporary variables for page length
*STORE 55 TO varpagel
*STORE 5 TO varlmarg
SET MARGIN TO varlmarg
* _PLENGTH=55
STORE " " TO printvar
STORE SPACE(60) TO namevar
STORE SPACE(60) TO filevar
filevar=HOME()+SPACE(LEN(filevar)-LEN(HOME()))
*******************************
SET PROCEDURE TO instsysr
CLEAR
DO startsrp
DEFINE POPUP prin_pop FROM 1,0
DEFINE BAR 1 OF prin_pop PROMPT "Print to Printer"
DEFINE BAR 2 OF prin_pop PROMPT "Print to Screen"
```

341

```
DEFINE BAR 3 OF prin_pop PROMPT "Print to ASCII File"
DEFINE BAR 4 OF prin_pop PROMPT "DONE!"
ON SELECTION POPUP prin_pop DO prin_pro
DEFINE POPUP sysr_pop FROM 1,0
DEFINE BAR 1 OF sysr_pop PROMPT "SRCs (TOE Listings)"
DEFINE BAR 2 OF sysr_pop PROMPT "SRC with LINs"
DEFINE BAR 3 OF sysr_pop PROMPT "Units"
DEFINE BAR 4 OF sysr_pop PROMPT "Units with LINs"
DEFINE BAR 5 OF sysr_pop PROMPT "Parameter Sets"
DEFINE BAR 6 OF sysr_pop PROMPT "Task Organizations"
DEFINE BAR 7 OF sysr_pop PROMPT "Task Organizations with Units"
DEFINE BAR 8 OF sysr_pop PROMPT "Orders"
DEFINE BAR 9 OF sysr_pop PROMPT "Orders with Task Organizations and Profiles"
DEFINE BAR 10 OF sysr_pop PROMPT "Orders with Task Orgs, Units, and Profiles"
DEFINE BAR 11 OF sysr_pop PROMPT "Change Printer Defaults"
DEFINE BAR 12 OF sysr_pop PROMPT "Return to Main Menu"
ON SELECTION POPUP sysr_pop DO sysr_pro
ON BAR 1 OF sysr_pop DO sysr_bar1
ON BAR 2 OF sysr_pop DO sysr_bar2
ON BAR 3 OF sysr_pop DO sysr_bar3
ON BAR 4 OF sysr_pop DO sysr_bar4
ON BAR 5 OF sysr_pop DO sysr_bar5
ON BAR 6 OF sysr_pop DO sysr_bar6
ON BAR 7 OF sysr_pop DO sysr_bar7
ON BAR 8 OF sysr_pop DO sysr_bar8
ON BAR 9 OF sysr_pop DO sysr_bar9
ON BAR 10 OF sysr_pop DO sysr_ba10
ON BAR 11 OF sysr_pop DO sysr_ba11
ON BAR 12 OF sysr_pop DO sysr_ba12
DO WHILE PROMPT()<>"Return to Main Menu"
    ACTIVATE POPUP sysr_pop
ENDDO (PROMPT()<>"Return to Main Menu")
*!***************************************************************
*!
*!        Procedure: SYSR_PRO
*!
*!        Called by: SYSREP.PRG
*!
*!            Calls: SRCRPT              (procedure in SYSREP.PRG)
*!                 : SRCLINR             (procedure in SYSREP.PRG)
*!                 : UNITRPT             (procedure in SYSREP.PRG)
*!                 : UNITLINR            (procedure in SYSREP.PRG)
*!                 : PROFILER            (procedure in SYSREP.PRG)
*!                 : TASKORGR            (procedure in SYSREP.PRG)
*!                 : TASKUNIT            (procedure in SYSREP.PRG)
*!                 : ORDERRPT            (procedure in SYSREP.PRG)
*!                 : ORDERTAS            (procedure in SYSREP.PRG)
*!                 : ORDTASUN            (procedure in SYSREP.PRG)
*!                 : PRNSETUP.PRG
*!                 : SYSREND             (procedure in SYSREP.PRG)
*!
*!***************************************************************
PROCEDURE sysr_pro
DO CASE
    * SRC report
CASE BAR() = 1
    DO srcrpt
    * SRC with LINs report
CASE BAR() = 2
    DO srclinr
```

342

```
          * Unit report
CASE BAR() = 3
     DO unitrpt
          * Unit with LINs report
CASE BAR() = 4
     DO unitlinr
          * Profile report
CASE BAR() = 5
     DO profiler
          * Task Organizations report
CASE BAR() = 6
     DO taskorgr
          * Task Organization with Units report
CASE BAR() = 7
     DO taskunit
          * Orders Report
CASE BAR() = 8
     DO orderrpt
          * Order with Task Orgs and Profiles report
CASE BAR() = 9
     DO ordertas
          * Order with Task Orgs, Profiles, and Units report
CASE BAR() = 10
     DO ordtasun
          * Change Printer Defaults
CASE BAR() = 11
     DO prnsetup
          * DONE Save SRC and end
CASE BAR() = 12
     DO sysrend
ENDCASE
* SRC report
*!********************************************************************
*!
*!        Procedure: SRCRPT
*!
*!        Called by: SYSR_PRO            (procedure in SYSREP.PRG)
*!
*!            Calls: SRCLST              (procedure in SYSREP.PRG)
*!
*!      Other Files: &FILEVAR
*!
*!********************************************************************
PROCEDURE srcrpt
CLEAR
DO WHILE PROMPT()<>"DONE!"
     ACTIVATE POPUP prin_pop
     IF PROMPT()<>"DONE!"
         SELECT 6
         SET ORDER TO src
         GO TOP
         DO CASE
         CASE PROMPT()= "Print to Printer"
             printdes="p"
             SET DEVICE TO PRINT
             SET PRINT ON
             ACTIVATE WINDOW process
             GO TOP
             DO srclst
             DEACTIVATE WINDOW process
```

343

```
                  SET PRINT OFF
                  SET DEVICE TO SCREEN
             CASE PROMPT()= "Print to Screen"
                  printdes="s"
                  GO TOP
                  DO srclst
             CASE PROMPT()= "Print to ASCII File"
                  printdes="f"
                  ACTIVATE WINDOW process
                  @ 1,1 SAY "Enter path and file name."
                  @ 3,1 GET filevar
                  READ
                  SET ALTERNATE TO &filevar
                  SET ALTERNATE ON
                  GO TOP
                  DO srclst
                  DEACTIVATE WINDOW process
                  SET ALTERNATE OFF
             CASE PROMPT()= "DONE!"
                  CLOSE DATA
                  DEACTIVATE POPUP
             ENDCASE
        ENDIF (PROMPT()<>"DONE!")
ENDDO (PROMPT()<>"DONE!")
*!*************************************************************
*!
*!      Procedure: SRCLST
*!
*!      Called by: SRCRPT               (procedure in SYSREP.PRG)
*!
*!*************************************************************
PROCEDURE srclst
CLEAR
? "SRC         UNIT DESIGNATION                    STRENGTH"
?
DO WHILE .NOT. EOF()
    ? src+" "+unit_desig+" "+TRANSFORM(strength,"@R 999,999")
    SKIP
    IF printdes="s" .AND. ROW()>20
        ?
        WAIT
        CLEAR
        ? "SRC         UNIT DESIGNATION                    STRENGTH"
        ?
    ENDIF (printdes="s" .AND. ROW()>20)
    IF printdes="p" .AND. PROW()>varpagel
        EJECT
        ? "SRC         UNIT DESIGNATION                    STRENGTH"
        ?
    ENDIF (printdes="p" .AND. PROW()>varpagel)
ENDDO (.NOT. EOF())
DO CASE
CASE printdes="s"
    ?
    WAIT
    CLEAR
CASE printdes="f"
    ?
    ? "END OF FILE"
CASE printdes="p"
```

```
    EJECT
ENDCASE
* SRC with LINs report
*!*********************************************************************
*!
*!        Procedure: SRCLINR
*!
*!        Called by: SYSR_PRO            (procedure in SYSREP.PRG)
*!
*!            Calls: GETSRC              (procedure in SYSREP.PRG)
*!                 : LINLST              (procedure in SYSREP.PRG)
*!
*!      Other Files: &FILEVAR
*!
*!*********************************************************************
PROCEDURE srclinr
* Choose SRC
CLEAR
DO getsrc
SELECT 5
SET FILTER TO src=srclist->src
GO TOP
* Choose where to print to
DO WHILE PROMPT()<>"DONE!"
   ACTIVATE POPUP prin_pop
   IF PROMPT()<>"DONE!"
      DO CASE
      CASE PROMPT()= "Print to Printer"
         printdes="p"
         SET DEVICE TO PRINT
         SET PRINT ON
         ACTIVATE WINDOW process
         GO TOP
         DO linlst
         DEACTIVATE WINDOW process
         SET PRINT OFF
         SET DEVICE TO SCREEN
      CASE PROMPT()= "Print to Screen"
         printdes="s"
         GO TOP
         DO linlst
      CASE PROMPT()= "Print to ASCII File"
         printdes="f"
         ACTIVATE WINDOW process
         @ 1,1 SAY "Enter path and file name."
         @ 3,1 GET filevar
         READ
         SET ALTERNATE TO &filevar
         SET ALTERNATE ON
         GO TOP
         DO linlst
         DEACTIVATE WINDOW process
         SET ALTERNATE OFF
      CASE PROMPT()= "DONE!"
         CLOSE DATA
         DEACTIVATE POPUP
      ENDCASE
   ENDIF (PROMPT()<>"DONE!")
ENDDO (PROMPT()<>"DONE!")
* SRC DATA
```

```
* LINs
*!*********************************************************************
*!
*!      Procedure: LINLST
*!
*!      Called by: SRCLINR            (procedure in SYSREP.PRG)
*!
*!*********************************************************************
PROCEDURE linlst
CLEAR
? "SRC          UNIT DESIGNATION                        STRENGTH"
?
? srclist->src+" "+srclist->unit_desig+" "+TRANSFORM(srclist->strength,"@R
999,999")
?
? "LIN      NOMENCLATURE                               QUANTITY"
?
DO WHILE .NOT. EOF()
   ? lin+"   "
   DO CASE
   CASE polrate->lin_nomen<>" "
      ?? polrate->lin_nomen
   CASE polrate->lin_nomen=" " .AND. ammorate->lin_nomen<>" "
      ?? ammorate->lin_nomen
   CASE polrate->lin_nomen=" " .AND. ammorate->lin_nomen=" "
      ?? "UNKNOWN                              "
   ENDCASE
   ?? " "+TRANSFORM(quantity,"@R 999,999")
   SKIP
   IF printdes="s" .AND. ROW()>20
      ?
      WAIT
      CLEAR
      ? "SRC          UNIT DESIGNATION                        STRENGTH"
      ?
      ? srclist->src+" "+srclist->unit_desig+"
"+TRANSFORM(srclist->strength,"@R 999,999")
      ?
      ? "LIN      NOMENCLATURE                               QUANTITY"
      ?
   ENDIF (printdes="s" .AND. ROW()>20)
   IF printdes="p" .AND. PROW()>varpagel
      EJECT
      ? "SRC          UNIT DESIGNATION                        STRENGTH"
      ?
      ? srclist->src+" "+srclist->unit_desig+"
"+TRANSFORM(srclist->strength,"@R 999,999")
      ?
      ? "LIN      NOMENCLATURE                               QUANTITY"
      ?
   ENDIF (printdes="p" .AND. PROW()>varpagel)
ENDDO (.NOT. EOF())
DO CASE
CASE printdes="s"
   ?
   WAIT
   CLEAR
CASE printdes="f"
   ?
   ? "END OF FILE"
```

```
CASE printdes="p"
   EJECT
ENDCASE
* Unit report
*!*********************************************************************
*!
*!       Procedure: UNITRPT
*!
*!      Called by: SYSR_PRO            (procedure in SYSREP.PRG)
*!
*!          Calls: UNITLST             (procedure in SYSREP.PRG)
*!
*!     Other Files: &FILEVAR
*!
*!*********************************************************************
PROCEDURE unitrpt
CLEAR
DO WHILE PROMPT()<>"DONE!"
   ACTIVATE POPUP prin_pop
   IF PROMPT()<>"DONE!"
      SELECT 2
      SET ORDER TO unitname
      GO TOP
      DO CASE
      CASE PROMPT()= "Print to Printer"
         printdes="p"
         SET DEVICE TO PRINT
         SET PRINT ON
         ACTIVATE WINDOW process
         GO TOP
         DO unitlst
         DEACTIVATE WINDOW process
         SET PRINT OFF
         SET DEVICE TO SCREEN
      CASE PROMPT()= "Print to Screen"
         printdes="s"
         GO TOP
         DO unitlst
      CASE PROMPT()= "Print to ASCII File"
         printdes="f"
         ACTIVATE WINDOW process
         @ 1,1 SAY "Enter path and file name."
         @ 3,1 GET filevar
         READ
         SET ALTERNATE TO &filevar
         SET ALTERNATE ON
         GO TOP
         DO unitlst
         DEACTIVATE WINDOW process
         SET ALTERNATE OFF
      CASE PROMPT()= "DONE!"
         CLOSE DATA
         DEACTIVATE POPUP
      ENDCASE
   ENDIF (PROMPT()<>"DONE!")
ENDDO (PROMPT()<>"DONE!")
* Unit with LINs report
*!*********************************************************************
*!
*!       Procedure: UNITLINR
```

347

```
*!
*!        Called by: SYSR_PRO          (procedure in SYSREP.PRG)
*!
*!            Calls: UNITLIN           (procedure in SYSREP.PRG)
*!
*!      Other Files: &FILEVAR
*!
*!***********************************************************************
PROCEDURE unitlinr
CLEAR
DO WHILE PROMPT()<>"DONE!"
   ACTIVATE POPUP prin_pop
   IF PROMPT()<>"DONE!"
      SELECT 2
      SET ORDER TO unitname
      GO TOP
      DO CASE
      CASE PROMPT()= "Print to Printer"
         printdes="p"
         SET DEVICE TO PRINT
         SET PRINT ON
         ACTIVATE WINDOW process
         GO TOP
         DO unitlin
         DEACTIVATE WINDOW process
         SET PRINT OFF
         SET DEVICE TO SCREEN
      CASE PROMPT()= "Print to Screen"
         printdes="s"
         GO TOP
         DO unitlin
      CASE PROMPT()= "Print to ASCII File"
         printdes="f"
         ACTIVATE WINDOW process
         @ 1,1 SAY "Enter path and file name."
         @ 3,1 GET filevar
         READ
         SET ALTERNATE TO &filevar
         SET ALTERNATE ON
         GO TOP
         DO unitlin
         DEACTIVATE WINDOW process
         SET ALTERNATE OFF
      CASE PROMPT()= "DONE!"
         CLOSE DATA
         DEACTIVATE POPUP
      ENDCASE
   ENDIF (PROMPT()<>"DONE!")
ENDDO (PROMPT()<>"DONE!")
* Profile report
*!***********************************************************************
*!
*!        Procedure: PROFILER
*!
*!        Called by: SYSR_PRO          (procedure in SYSREP.PRG)
*!
*!            Calls: PROFLST           (procedure in SYSREP.PRG)
*!
*!      Other Files: &FILEVAR
*!
```

```
*!*****************************************************************
PROCEDURE profiler
CLEAR
DO WHILE PROMPT()<>"DONE!"
    ACTIVATE POPUP prin_pop
    IF PROMPT()<>"DONE!"
        SELECT 4
        SET ORDER TO profilenam
        GO TOP
        DO CASE
        CASE PROMPT()= "Print to Printer"
            printdes="p"
            SET DEVICE TO PRINT
            SET PRINT ON
            ACTIVATE WINDOW process
            GO TOP
            DO proflst
            DEACTIVATE WINDOW process
            SET PRINT OFF
            SET DEVICE TO SCREEN
        CASE PROMPT()= "Print to Screen"
            printdes="s"
            GO TOP
            DO proflst
        CASE PROMPT()= "Print to ASCII File"
            printdes="f"
            ACTIVATE WINDOW process
            @ 1,1 SAY "Enter path and file name."
            @ 3,1 GET filevar
            READ
            SET ALTERNATE TO &filevar
            SET ALTERNATE ON
            GO TOP
            DO proflst
            DEACTIVATE WINDOW process
            SET ALTERNATE OFF
        CASE PROMPT()= "DONE!"
            CLOSE DATA
            DEACTIVATE POPUP
        ENDCASE
    ENDIF (PROMPT()<>"DONE!")
ENDDO (PROMPT()<>"DONE!")
* Task Organizations report
*!*****************************************************************
*!
*!        Procedure: TASKORGR
*!
*!        Called by: SYSR_PRO            (procedure in SYSREP.PRG)
*!
*!            Calls: TASKLST             (procedure in SYSREP.PRG)
*!
*!      Other Files: &FILEVAR
*!
*!*****************************************************************
PROCEDURE taskorgr
CLEAR
DO WHILE PROMPT()<>"DONE!"
    ACTIVATE POPUP prin_pop
    IF PROMPT()<>"DONE!"
        SELECT 1
```

```
            SET ORDER TO uniqueta
            GO TOP
            DO CASE
            CASE PROMPT()= "Print to Printer"
                printdes="p"
                SET DEVICE TO PRINT
                SET PRINT ON
                ACTIVATE WINDOW process
                GO TOP
                DO tasklst
                DEACTIVATE WINDOW process
                SET PRINT OFF
                SET DEVICE TO SCREEN
            CASE PROMPT()= "Print to Screen"
                printdes="s"
                GO TOP
                DO tasklst
            CASE PROMPT()= "Print to ASCII File"
                printdes="f"
                ACTIVATE WINDOW process
                @ 1,1 SAY "Enter path and file name."
                @ 3,1 GET filevar          ,
                READ
                SET ALTERNATE TO &filevar
                SET ALTERNATE ON
                GO TOP
                DO tasklst
                DEACTIVATE WINDOW process
                SET ALTERNATE OFF
            CASE PROMPT()= "DONE!"
                CLOSE DATA
                DEACTIVATE POPUP
            ENDCASE
        ENDIF (PROMPT()<>"DONE!")
    ENDDO (PROMPT()<>"DONE!")
    * Task Organization with Units report
    *!*******************************************************************
    *!
    *!        Procedure: TASKUNIT
    *!
    *!        Called by: SYSR_PRO          (procedure in SYSREP.PRG)
    *!
    *!            Calls: TASKUNI            (procedure in SYSREP.PRG)
    *!
    *!      Other Files: &FILEVAR
    *!
    *!*******************************************************************
    PROCEDURE taskunit
    CLEAR
    DO WHILE PROMPT()<>"DONE!"
        ACTIVATE POPUP prin_pop
        IF PROMPT()<>"DONE!"
            SELECT 1
            SET ORDER TO taskorgnam
            GO TOP
            DO CASE
            CASE PROMPT()= "Print to Printer"
                printdes="p"
                SET DEVICE TO PRINT
                SET PRINT ON
```

350

```
                ACTIVATE WINDOW process
                GO TOP
                DO taskuni
                DEACTIVATE WINDOW process
                SET PRINT OFF
                SET DEVICE TO SCREEN
            CASE PROMPT()= "Print to Screen"
                printdes="s"
                GO TOP
                DO taskuni
            CASE PROMPT()= "Print to ASCII File"
                printdes="f"
                ACTIVATE WINDOW process
                @ 1,1 SAY "Enter path and file name."
                @ 3,1 GET filevar
                READ
                SET ALTERNATE TO &filevar
                SET ALTERNATE ON
                GO TOP
                DO taskuni
                DEACTIVATE WINDOW process
                SET ALTERNATE OFF
            CASE PROMPT()= "DONE!'"
                CLOSE DATA
                DEACTIVATE POPUP
            ENDCASE
        ENDIF (PROMPT()<>"DONE!")
ENDDO (PROMPT()<>"DONE!")
* Orders Report
*!*************************************************************************
*!
*!        Procedure: ORDERRPT
*!
*!        Called by: SYSR_PRO              (procedure in SYSREP.PRG)
*!
*!            Calls: ORDERLST              (procedure in SYSREP.PRG)
*!
*!      Other Files: &FILEVAR
*!
*!*************************************************************************
PROCEDURE orderrpt
CLEAR
DO WHILE PROMPT()<>"DONE!"
    ACTIVATE POPUP prin_pop
    IF PROMPT()<>"DONE!"
        SELECT 3
        SET ORDER TO NUMBER
        GO TOP
        DO CASE
        CASE PROMPT()= "Print to Printer"
            printdes="p"
            SET DEVICE TO PRINT
            SET PRINT ON
            ACTIVATE WINDOW process
            GO TOP
            DO orderlst
            DEACTIVATE WINDOW process
            SET PRINT OFF
            SET DEVICE TO SCREEN
        CASE PROMPT()= "Print to Screen"
```

351

```
            printdes="s"
            GO TOP
            DO orderlst
         CASE PROMPT()= "Print to ASCII File"
            printdes="f"
            ACTIVATE WINDOW process
            @ 1,1 SAY "Enter path and file name."
            @ 3,1 GET filevar
            READ
            SET ALTERNATE TO &filevar
            SET ALTERNATE ON
            GO TOP
            DO orderlst
            DEACTIVATE WINDOW process
            SET ALTERNATE OFF
         CASE PROMPT()= "DONE!"
            CLOSE DATA
            DEACTIVATE POPUP
         ENDCASE
      ENDIF (PROMPT()<>"DONE!")
   ENDDO (PROMPT()<>"DONE!")
   * Order with Task Orgs and Profiles report
   *!***********************************************************
   *!
   *!       Procedure: ORDERTAS
   *!
   *!       Called by: SYSR_PRO            (procedure in SYSREP.PRG)
   *!
   *!           Calls: ORDERTP             (procedure in SYSREP.PRG)
   *!
   *!      Other Files: &FILEVAR
   *!
   *!***********************************************************
   PROCEDURE ordertas
   CLEAR
   DO WHILE PROMPT()<>"DONE!"
      ACTIVATE POPUP prin_pop
      IF PROMPT()<>"DONE!"
         SELECT 3
         SET ORDER TO NUMBER
         GO TOP
         DO CASE
         CASE PROMPT()= "Print to Printer"
            printdes="p"
            SET DEVICE TO PRINT
            SET PRINT ON
            ACTIVATE WINDOW process
            GO TOP
            DO ordertp
            DEACTIVATE WINDOW process
            SET PRINT OFF
            SET DEVICE TO SCREEN
         CASE PROMPT()= "Print to Screen"
            printdes="s"
            GO TOP
            DO ordertp
         CASE PROMPT()= "Print to ASCII File"
            printdes="f"
            ACTIVATE WINDOW process
            @ 1,1 SAY "Enter path and file name."
```

```
            @ 3,1 GET filevar
            READ
            SET ALTERNATE TO &filevar
            SET ALTERNATE ON
            GO TOP
            DO ordertp
            DEACTIVATE WINDOW process
            SET ALTERNATE OFF
         CASE PROMPT()= "DONE!"
            CLOSE DATA
            DEACTIVATE POPUP
         ENDCASE
      ENDIF (PROMPT()<>"DONE!")
ENDDO (PROMPT()<>"DONE!")
* Order with Task Orgs, Profiles, and Units report
*!****************************************************************
*!
*!       Procedure: ORDTASUN
*!
*!       Called by: SYSR_PRO           (procedure in SYSREP.PRG)
*!
*!           Calls: ORDERTUP           (procedure in SYSREP.PRG)
*!
*!      Other Files: &FILEVAR
*!
*!****************************************************************
PROCEDURE ordtasun
CLEAR
DO WHILE PROMPT()<>"DONE!"
   ACTIVATE POPUP prin_pop
   IF PROMPT()<>"DONE!"
      SELECT 3
      SET ORDER TO NUMBER
      GO TOP
      DO CASE
      CASE PROMPT()= "Print to Printer"
         printdes="p"
         SET DEVICE TO PRINT
         SET PRINT ON
         ACTIVATE WINDOW process
         GO TOP
         DO ordertup
         DEACTIVATE WINDOW process
         SET PRINT OFF
         SET DEVICE TO SCREEN
      CASE PROMPT()= "Print to Screen"
         printdes="s"
         GO TOP
         DO ordertup
      CASE PROMPT()= "Print to ASCII File"
         printdes="f"
         ACTIVATE WINDOW process
         @ 1,1 SAY "Enter path and file name."
         @ 3,1 GET filevar
         READ
         SET ALTERNATE TO &filevar
         SET ALTERNATE ON
         GO TOP
         DO ordertup
         DEACTIVATE WINDOW process
```

353

```
            SET ALTERNATE OFF
         CASE PROMPT()= "DONE!"
            CLOSE DATA
            DEACTIVATE POPUP
         ENDCASE
     ENDIF (PROMPT()<>"DONE!")
ENDDO (PROMPT()<>"DONE!")
* Change Printer Defaults
* DO prnsetup is seperate program
* DONE Save SRC and end
*!****************************************************************
*!
*!        Procedure: SYSREND
*!
*!        Called by: SYSR_PRO             (procedure in SYSREP.PRG)
*!
*!****************************************************************
PROCEDURE sysrend
CLOSE DATA
DEACTIVATE POPUP
CLEAR
RETURN
*!****************************************************************
*!
*!        Procedure: STARTSRP
*!
*!        Called by: SYSREP.PRG
*!
*!            Calls: STARTHLP.PRG
*!
*!             Uses: TASKORG.DBF
*!                 : CSRCLIST.DBF
*!                 : ORDERLST.DBF
*!                 : CPROFILE.DBF
*!                 : SRCLINQT.DBF
*!                 : SRCLIST.DBF
*!                 : AMMORATE.DBF
*!                 : POLRATE.DBF
*!                 : CSRCLINQ.DBF
*!                 : ORDERTAS.DBF
*!
*!        MDX files: TASKORG.MDX
*!                 : CSRCLIST.MDX
*!                 : ORDERLST.MDX
*!                 : CPROFILE.MDX
*!                 : SRCLINQT.MDX
*!                 : SRCLIST.MDX
*!                 : AMMORATE.MDX
*!                 : POLRATE.MDX
*!                 : CSRCLINQ.MDX
*!                 : ORDERTAS.MDX
*!
*!****************************************************************
PROCEDURE startsrp
*Open Files
DO starthlp
SELECT 1
USE taskorg
SELECT 2
USE csrclist
```

354

```
SELECT 3
USE orderlst
SELECT 4
USE cprofile
SELECT 5
USE srclinqt
SELECT 6
USE srclist
SELECT 8
USE ammorate ORDER lin
SELECT 12
USE polrate ORDER lin
SELECT 14
USE csrclinq
SELECT 18
USE ordertas
* Set Relations
SELECT 6
SET ORDER TO src
SELECT 5
SET ORDER TO src
SET RELATION TO src INTO srclist
SET ORDER TO lin
SET RELATION TO lin INTO polrate, lin INTO ammorate
SELECT 14
SET ORDER TO lin
SET RELATION TO lin INTO polrate, lin INTO ammorate
SET ORDER TO unitname
SELECT 2
SET ORDER TO unitname
SET RELATION TO unitname INTO csrclinq
SELECT 1
SET ORDER TO unitname
SET RELATION TO unitname INTO csrclist
SET ORDER TO taskorgnam
SELECT 18
SET ORDER TO taskorgnam
SET RELATION TO taskorgnam INTO taskorg
SET ORDER TO NUMBER
SELECT 3
SET ORDER TO NUMBER
SET RELATION TO NUMBER INTO ordertas
*!****************************************************************
*!
*!      Procedure: GETSRC
*!
*!      Called by: SRCLINR            (procedure in SYSREP.PRG)
*!
*!          Calls: SETBROWS.PRG
*!               : ENDBROWS.PRG
*!               : DISPSRC            (procedure in SYSREP.PRG)
*!
*!****************************************************************
PROCEDURE getsrc
SELECT 6
GO TOP
*      SET MESSAGE TO "Choose an SRC, then Ctrl-W"
ACTIVATE WINDOW browser
DO setbrows
BROWSE NOEDIT NOAPPEND NOORGANIZE NODELETE COMPRESS
```

```
DO endbrows
DEACTIVATE WINDOW browser
DO dispsrc
statuslv=.y.
KEYBOARD "{DNARROW}"
*!***************************************************************
*!
*!       Procedure: DISPSRC
*!
*!       Called by: GETSRC              (procedure in SYSREP.PRG)
*!
*!***************************************************************
PROCEDURE dispsrc
@ 18,5 SAY "      SRC: "+srclist->src+"  Strength: "+;
   LTRIM(STR(srclist->strength))
@ 19,5 SAY "     Unit: "+srclist->unit_desig
*!***************************************************************
*!
*!       Procedure: DISPPROF
*!
*!       Called by: NEWPROFI            (procedure in CREATEPR.PRG)
*!                : VIEWPROF            (procedure in CREATEPR.PRG)
*!                : SELEPROF            (procedure in EDITPROF.PRG)
*!                : CHANGEAP            (procedure in EDITPROF.PRG)
*!                : CHANGEAC            (procedure in EDITPROF.PRG)
*!                : CHANGEI             (procedure in EDITPROF.PRG)
*!                : CHANGEIF            (procedure in EDITPROF.PRG)
*!                : CHANGECI            (procedure in EDITPROF.PRG)
*!                : CHANGEH2            (procedure in EDITPROF.PRG)
*!                : CHANGEME            (procedure in EDITPROF.PRG)
*!                : SELECTPRO           (procedure in DELPROF.PRG)
*!                : FINDPRO             (procedure in CREATEOR.PRG)
*!
*!***************************************************************
PROCEDURE dispprof
@ 20,5 SAY "  Profile: "+profiles->profile
*!***************************************************************
*!
*!       Procedure: LISTSRC
*!
*!***************************************************************
PROCEDURE listsrc
? "       SRC: "+srclist->src+"  Strength: "+;
   LTRIM(STR(srclist->strength))
? "      Unit: "+srclist->unit_desig
*!***************************************************************
*!
*!       Procedure: LISTPROF
*!
*!***************************************************************
PROCEDURE listprof
? "  Profile: "+profiles->profile
*!***************************************************************
*!
*!       Procedure: PRIN_PRO
*!
*!       Called by: SYSREP.PRG
*!                : PRINTPOL.PRG
*!                : PRINTAMM.PRG
*!                : PRINTOTH.PRG
```

356

```
*!
*!           Calls:- PRNSETUP.PRG                                  '
*!                 : STARTPLR          (procedure in PRINTPOL.PRG)
*!                 : REPORTHD          (procedure in PRINTPOL.PRG)
*!                 : TFHEAD            (procedure in PRINTPOL.PRG)
*!                 : LINPOL            (procedure in PRINTPOL.PRG)
*!                 : POLCALC           (procedure in PRINTPOL.PRG)
*!                 : PUNITPOL          (procedure in PRINTPOL.PRG)
*!                 : PTASKPOL          (procedure in PRINTPOL.PRG)
*!                 : PORDEPOL          (procedure in PRINTPOL.PRG)
*!                 : LINAMM            (procedure in PRINTAMM.PRG)
*!                 : AMMOCALC          (procedure in PRINTAMM.PRG)
*!                 : PUNITAMM          (procedure in PRINTAMM.PRG)
*!                 : PTASKAMM          (procedure in PRINTAMM.PRG)
*!                 : PORDEAMM          (procedure in PRINTAMM.PRG)
*!                 : OTHECALC          (procedure in PRINTOTH.PRG)
*!                 : PUNITOTH          (procedure in PRINTOTH.PRG)
*!                 : PTASKOTH          (procedure in PRINTOTH.PRG)
*!                 : PORDEOTH          (procedure in PRINTOTH.PRG)
*!
*!*******************************************************************
PROCEDURE prin_pro
DO CASE
CASE PROMPT()="Print to Printer"
   printvar="p"
   SET PRINT ON
   SET DEVICE TO PRINT
   DEACTIVATE POPUP
CASE PROMPT()="Print to Screen"
   printvar="s"
   SET DEVICE TO SCREEN
   DEACTIVATE POPUP
CASE PROMPT()="Print to ASCII File"
   printvar="f"
   DEACTIVATE POPUP
CASE PROMPT()="DONE!"
   DEACTIVATE POPUP
ENDCASE
*!*******************************************************************
*!
*!      Procedure: SRCLST
*!
*!      Called by: SRCRPT               (procedure in SYSREP.PRG)
*!
*!*******************************************************************
PROCEDURE srclst
CLEAR
? "SRC          UNIT DESIGNATION                      STRENGTH"
?
DO WHILE .NOT. EOF()
   ? src+" "+unit_desig+" "+TRANSFORM(strength,"@R 999,999")
   SKIP
   IF printdes="s" .AND. ROW()>20
      ?
      WAIT
      CLEAR
      ? "SRC          UNIT DESIGNATION                      STRENGTH"
      ?
   ENDIF (printdes="s" .AND. ROW()>20)
   IF printdes="p" .AND. PROW()>varpagel
```
357

```
        EJECT
        ? "SRC        UNIT DESIGNATION            '                    STRENGTH"
        ?
    ENDIF (printdes="p" .AND. PROW()>varpagel)
ENDDO (.NOT. EOF())
DO CASE
CASE printdes="s"
    ?
    WAIT
    CLEAR
CASE printdes="f"
    ?
    ? "END OF FILE"
CASE printdes="p"
    EJECT
ENDCASE
*!******************************************************************
*!
*!      Procedure: UNITLST
*!
*!      Called by: UNITRPT              (procedure in SYSREP.PRG)
*!
*!******************************************************************
PROCEDURE unitlst
CLEAR
? "UNIT                                          STRENGTH
ECHELON"
?
DO WHILE .NOT. EOF()
    ? unitname+" "+TRANSFORM(strength,"@R 999,999")+" "+echelon
    SKIP
    IF printdes="s" .AND. ROW()>20
        ?
        WAIT
        CLEAR
        ? "SRC        UNIT DESIGNATION                    STRENGTH"
        ?
    ENDIF (printdes="s" .AND. ROW()>20)
    IF printdes="p" .AND. PROW()>varpagel
        EJECT
        ? "SRC        UNIT DESIGNATION                    STRENGTH"
        ?
    ENDIF (printdes="p" .AND. PROW()>varpagel)
ENDDO (.NOT. EOF())
DO CASE
CASE printdes="s"
    ?
    WAIT
    CLEAR
CASE printdes="f"
    ?
    ? "END OF FILE"
CASE printdes="p"
    EJECT
ENDCASE
*!******************************************************************
*!
*!      Procedure: UNITLIN
*!
*!      Called by: UNITLINR             (procedure in SYSREP.PRG)
```

358

```
*!
*!*************************************************************
PROCEDURE unitlin
CLEAR
? "UNIT                                              STRENGTH
ECHELON"
?
DO WHILE .NOT. EOF()
    ? unitname+" "+TRANSFORM(strength,"@R 999,999")+" "+echelon
    ?
    SELECT 14
    SET ORDER TO unitname
    GO TOP
    SEEK csrclist->unitname
    IF FOUND()
        ? "LIN      QTY     NOMENCLATURE"
        DO WHILE csrclinq->unitname=csrclist->unitname
            ? lin+" "+TRANSFORM(quantity,"999,999")+" "
            IF ammorate->lin<>" "
                ?? ammorate->lin_nomen
            ELSE
                IF polrate->lin_nomen<>" "
                    ?? polrate->lin_nomen
                ENDIF (polrate->lin_nomen<>" ")
            ENDIF (ammorate->lin<>" ")
            SKIP
            IF printdes="s" .AND. ROW()>20
                ?
                WAIT
                CLEAR
                ? "LIN      QTY     NOMENCLATURE"
                ?
            ENDIF (printdes="s" .AND. ROW()>20)
            IF printdes="p" .AND. PROW()>varpagel
                EJECT
                ? "LIN      QTY     NOMENCLATURE"
                ?
            ENDIF (printdes="p" .AND. PROW()>varpagel)
        ENDDO (csrclinq->unitname=csrclist->unitname)
    ENDIF (FOUND())
    ?
    ?
    SELECT 2
    SKIP
    IF printdes="s" .AND. ROW()>20
        ?
        WAIT
        CLEAR
        ? "SRC        UNIT DESIGNATION                    STRENGTH"
        ?
    ENDIF (printdes="s" .AND. ROW()>20)
    IF printdes="p" .AND. PROW()>varpagel
        EJECT
        ? "SRC        UNIT DESIGNATION                    STRENGTH"
        ?
    ENDIF (printdes="p" .AND. PROW()>varpagel)
ENDDO (.NOT. EOF())
DO CASE
CASE printdes="s"
    ?
```

```
    WAIT
    CLEAR
CASE printdes="f"
    ?
    ? "END OF FILE"
CASE printdes="p"
    EJECT
ENDCASE
*!******************************************************************
*!
*!      Procedure: PROFLST
*!
*!      Called by: PROFILER          (procedure in SYSREP.PRG)
*!
*!******************************************************************
PROCEDURE proflst
CLEAR
DO WHILE .NOT. EOF()
    ? "PARAMETER SET:    "+profilenam
    ? "  DESCRIPTION:    "+profiledes
    ? "  AREA PROFILE:   "+profile
    ? "  INTENSITY:      "+INTENSITY+"  "+"FACTOR: "+TRANSFORM(factor,"999.99")
    ? "  POSTURE:        "+posture
    ? "  H20 USE:        "+water_name
    ? "  CLASS I:        "+cl_i_1+" / "+cl_i_2+" / "+cl_i_3
    ? "  CLASS VIII:     "+med_name
    ?
    SKIP
    IF printdes="s" .AND. ROW()>20
        ?
        WAIT
    ENDIF (printdes="s" .AND. ROW()>20)
    IF printdes="p" .AND. PROW()>varpagel
        EJECT
    ENDIF (printdes="p" .AND. PROW()>varpagel)
ENDDO (.NOT. EOF())
DO CASE
CASE printdes="s"
    ?
    WAIT
    CLEAR
CASE printdes="f"
    ?
    ? "END OF FILE"
CASE printdes="p"
    EJECT
ENDCASE
*!******************************************************************
*!
*!      Procedure: TASKLST
*!
*!      Called by: TASKORGR          (procedure in SYSREP.PRG)
*!
*!******************************************************************
PROCEDURE tasklst
CLEAR
? "TASK ORGANIZATIONS:"
?
DO WHILE .NOT. EOF()
    ? taskorgnam
```

360

```
    ?
    SKIP
    IF printdes="s" .AND. ROW()>20
        ?
        WAIT
    ENDIF (printdes="s" .AND. ROW()>20)
    IF printdes="p" .AND. PROW()>varpagel
        EJECT
    ENDIF (printdes="p" .AND. PROW()>varpagel)
ENDDO (.NOT. EOF())
DO CASE
CASE printdes="s"
    ?
    WAIT
    CLEAR
CASE printdes="f"
    ?
    ? "END OF FILE"
CASE printdes="p"
    EJECT
ENDCASE
*!*******************************************************************
*!
*!      Procedure: TASKUNI
*!
*!      Called by: TASKUNIT              (procedure in SYSREP.PRG)
*!
*!*******************************************************************
PROCEDURE taskuni
CLEAR
? "TASK ORGANIZATIONS:"
?
DO WHILE .NOT. EOF()
    ? taskorgnam
    namevar=taskorgnam
    ?
    DO WHILE .NOT. EOF()
        ? "  "+unitname
        IF printdes="s" .AND. ROW()>20
            ?
            WAIT
        ENDIF (printdes="s" .AND. ROW()>20)
        IF printdes="p" .AND. PROW()>varpagel
            EJECT
        ENDIF (printdes="p" .AND. PROW()>varpagel)
        SKIP
        IF taskorgnam<>namevar
            ?
            ?
            ? taskorgnam
            namevar=taskorgnam
            ?
        ENDIF (taskorgnam<>namevar)
    ENDDO (.NOT. EOF())
ENDDO (.NOT. EOF())
DO CASE
CASE printdes="s"
    ?
    WAIT
    CLEAR
```

361

```
CASE printdes="f"
    ?
    ? "END OF FILE"
CASE printdes="p"
    EJECT
ENDCASE
*!*********************************************************************
*!
*!       Procedure: ORDERTP
*!
*!       Called by: ORDERTAS              (procedure in SYSREP.PRG)
*!
*!*********************************************************************
PROCEDURE ordertp
CLEAR
? "ORDERS:"
?
DO WHILE .NOT. EOF()
    ? NUMBER+" "+name
    ? descriptio
    ? "PHASES: "+TRANSFORM(phases,"9")+" Phase Lengths: "
    IF phases>0
        ?? "1st: "+TRANSFORM(phase1len,"99.99")+"   "
    ENDIF (phases>0)
    IF phases>1
        ?? "2nd: "+TRANSFORM(phase2len,"99.99.")+"   "
    ENDIF (phases>1)
    IF phases>2
        ?? "3rd: "+TRANSFORM(phase3len,"99.99")+"   "
    ENDIF (phases>2)
    IF phases>3
        ?? "4th: "+TRANSFORM(phase4len,"99.99")+"   "
    ENDIF (phases>3)
    IF phases>4
        ?
        ?? "5th: "+TRANSFORM(phase5len,"99.99")+"   "
    ENDIF (phases>4)
    IF phases>5
        ?? "6th: "+TRANSFORM(phase6len,"99.99")+"   "
    ENDIF (phases>5)
    IF phases>6
        ?? "7th: "+TRANSFORM(phase7len,"99.99")+"   "
    ENDIF (phases>6)
    IF phases>7
        ?? "8th: "+TRANSFORM(phase8len,"99.99")+"   "
    ENDIF (phases>7)
    IF phases>8
        ?? "9th: "+TRANSFORM(phase9len,"99.99")
    ENDIF (phases>8)
    ?
    * Task Orgs & Phases
    SELECT 18
    SET ORDER TO NUMBER
    SEEK orderlst->number
    IF FOUND()
        SET ORDER TO uniquetn
        ? "TASK ORGANIZATIONS"
        DO WHILE NUMBER=orderlst->number
            ? "   "+taskorgnam
            SKIP
```

362

```
        ENDDO (NUMBER=orderlst->number)
        ?
        IF printdes="s" .AND. ROW()>20
            ?
            WAIT
        ENDIF (printdes="s" .AND. ROW()>20)
        IF printdes="p" .AND. PROW()>varpagel
            EJECT
        ENDIF (printdes="p" .AND. PROW()>varpagel)
    ENDIF (FOUND())
    SELECT 3
    SKIP
    IF printdes="s" .AND. ROW()>20
        ?
        WAIT
    ENDIF (printdes="s" .AND. ROW()>20)
    IF printdes="p" .AND. PROW()>varpagel
        EJECT
    ENDIF (printdes="p" .AND. PROW()>varpagel)
ENDDO (.NOT. EOF())
DO CASE
CASE printdes="s"
    ?
    WAIT
    CLEAR
CASE printdes="f"
    ?
    ? "END OF FILE"
CASE printdes="p"
    EJECT
ENDCASE
*!******************************************************************
*!
*!      Procedure: ORDERLST
*!
*!      Called by: ORDERRPT              (procedure in SYSREP.PRG)
*!
*!******************************************************************
PROCEDURE orderlst
CLEAR
? "ORDERS:"
?
DO WHILE .NOT. EOF()
    ? NUMBER+" "+name
    ? descriptio
    ? "PHASES: "+TRANSFORM(phases,"9")+" Phase Lengths: "
    IF phases>0
        ?? "1st: "+TRANSFORM(phase1len,"99.99")+"   "
    ENDIF (phases>0)
    IF phases>1
        ?? "2nd: "+TRANSFORM(phase2len,"99.99.")+"   "
    ENDIF (phases>1)
    IF phases>2
        ?? "3rd: "+TRANSFORM(phase3len,"99.99")+"   "
    ENDIF (phases>2)
    IF phases>3
        ?? "4th: "+TRANSFORM(phase4len,"99.99")+"   "
    ENDIF (phases>3)
    IF phases>4
        ?
```

```
        ?? "5th: "+TRANSFORM(phase5len,"99.99")+"    "
    ENDIF (phases>4)
    IF phases>5
        ?? "6th: "+TRANSFORM(phase6len,"99.99")+"    "
    ENDIF (phases>5)
    IF phases>6
        ?? "7th: "+TRANSFORM(phase7len,"99.99")+"    "
    ENDIF (phases>6)
    IF phases>7
        ?? "8th: "+TRANSFORM(phase8len,"99.99")+"    "
    ENDIF (phases>7)
    IF phases>8
        ?? "9th: "+TRANSFORM(phase9len,"99.99")
    ENDIF (phases>8)
    ?
    SKIP
    IF printdes="s" .AND. ROW()>20
        ?
        WAIT
    ENDIF (printdes="s" .AND. ROW()>20)
    IF printdes="p" .AND. PROW()>varpagel
        EJECT
    ENDIF (printdes="p" .AND. PROW()>varpagel)
ENDDO (.NOT. EOF())
DO CASE
CASE printdes="s"
    ?
    WAIT
    CLEAR
CASE printdes="f"
    ?
    ? "END OF FILE"
CASE printdes="p"
    EJECT
ENDCASE
*!**********************************************************************
*!
*!      Procedure: ORDERTUP
*!
*!      Called by: ORDTASUN              (procedure in SYSREP.PRG)
*!
*!**********************************************************************
PROCEDURE ordertup
CLEAR
? "ORDERS:"
?
DO WHILE .NOT. EOF()
    ? NUMBER+" "+name
    ? descriptio
    ? "PHASES: "+TRANSFORM(phases,"9")+" Phase Lengths: "
    IF phases>0
        ?? "1st: "+TRANSFORM(phase1len,"99.99")+"    "
    ENDIF (phases>0)
    IF phases>1
        ?? "2nd: "+TRANSFORM(phase2len,"99.99.")+"    "
    ENDIF (phases>1)
    IF phases>2
        ?? "3rd: "+TRANSFORM(phase3len,"99.99")+"    "
    ENDIF (phases>2)
    IF phases>3
```

364

```
    ?? "4th: "+TRANSFORM(phase4len,"99.99")+"    "
ENDIF (phases>3)
IF phases>4
    ?
    ?? "5th: "+TRANSFORM(phase5len,"99.99")+"    "
ENDIF (phases>4)
IF phases>5
    ?? "6th: "+TRANSFORM(phase6len,"99.99")+"    "
ENDIF (phases>5)
IF phases>6
    ?? "7th: "+TRANSFORM(phase7len,"99.99")+"    "
ENDIF (phases>6)
IF phases>7
    ?? "8th: "+TRANSFORM(phase8len,"99.99")+"    "
ENDIF (phases>7)
IF phases>8
    ?? "9th: "+TRANSFORM(phase9len,"99.99")
ENDIF (phases>8)
?
* Task Orgs & Phases
SELECT 18
SET ORDER TO NUMBER
SEEK orderlst->number
IF FOUND()
    SET ORDER TO uniquetn
    ? "TASK ORGANIZATIONS"
    DO WHILE NUMBER=orderlst->number
        ? "   "+taskorgnam
        SELECT 1
        SET ORDER TO unitname
        SET FILTER TO taskorgnam=ordertas->taskorgnam
        GO TOP
        DO WHILE .NOT. EOF()
            ? "      "+unitname
            IF printdes="s" .AND. ROW()>20
                ?
                WAIT
            ENDIF (printdes="s" .AND. ROW()>20)
            IF printdes="p" .AND. PROW()>varpagel
                EJECT
            ENDIF (printdes="p" .AND. PROW()>varpagel)
            SKIP
        ENDDO (.NOT. EOF())
        SET FILTER TO
        SELECT 18
        SKIP
    ENDDO (NUMBER=orderlst->number)
    ?
    IF printdes="s" .AND. ROW()>20
        ?
        WAIT
    ENDIF (printdes="s" .AND. ROW()>20)
    IF printdes="p" .AND. PROW()>varpagel
        EJECT
    ENDIF (printdes="p" .AND. PROW()>varpagel)
ENDIF (FOUND())
SELECT 3
SKIP
IF printdes="s" .AND. ROW()>20
    ?
```

```
        WAIT
    ENDIF (printdes="s" .AND. ROW()>20)
    IF printdes="p" .AND. PROW()>varpagel
        EJECT
    ENDIF (printdes="p" .AND. PROW()>varpagel)
ENDDO (.NOT. EOF())
DO CASE
CASE printdes="s"
    ?
    WAIT
    CLEAR
CASE printdes="f"
    ?
    ? "END OF FILE"
CASE printdes="p"
    EJECT
ENDCASE

*: EOF: SYSREP.PRG
```

```
*:******************************************************************
*:
*:          Program: D:\SURE\UNCOMPRE.PRG
*:
*:          Called by: UTIL_PRO            (procedure in SURE.PRG)
*:
*:          Documented 04/16/94 at 11:02            SNAP!  version 5.00
*:******************************************************************
STORE SPACE(10) TO printvar
ACTIVATE WINDOW small
@ 1,1 SAY "Select printer type with SPACE"
@ 2,1 SAY "and RETURN to select."
@ 3,1 GET printvar;
   PICTURE "@M Dot Matrix,Laser,Do not set!";
   MESSAGE "Dot Matrix for Epson/IBM compatibles, Laser for HP Laser Printers"
READ
DO CASE
CASE printvar="Dot Matrix"
   ??? CHR(27)+CHR(18)
CASE printvar="Laser"
   ??? CHR(27)+"@10O"
   ??? CHR(27)+"(8U"
   ??? CHR(27)+"(s0p10h12v0s0b3T"
ENDCASE
DEACTIVATE WINDOW small
RETURN
RETURN
*: EOF: UNCOMPRE.PRG
```

ANNEX E

PROGRAMMERS REFERENCE

Table of Contents
Programmer's Reference
for SURE 2.0

```
                    System: SURE 2.0
                    System Summary

----------------------------------------------------------------
This system has:
 16386 lines of code
    18 program files
    35 procedure files
   321 procedures and functions
    29 databases
    24 multiple index files
    34 index files
     0 report forms
     0 format files
     0 label forms
     0 binary files
     1 memory variable file
     0 menu files
     0 screen files
     1 other file
     2 cross-referenced tokens
```

Programs and procedures:

ADDDATA	(procedure in CREATESR.PRG)
ADDLINCA	(procedure in EDITUNIT.PRG)
BACKDBF.PRG	
CHANGEEH	(procedure in EDITUNIT.PRG)
CHANGELI	(procedure in EDITUNIT.PRG)
CLOSEALL.PRG	
COMPRESS.PRG	
COPYCSRC	(procedure in CREATESR.PRG)
COPYLINS	(procedure in CREATESR.PRG)
COPYSRC	(procedure in CREATESR.PRG)
CSRCEND	(procedure in CREATESR.PRG)
CSRCENDA	(procedure in CREATESR.PRG)
CSRC_BAR1	(procedure in INSTUNIT.PRG)
CSRC_BAR2	(procedure in INSTUNIT.PRG)
CSRC_BAR3	(procedure in INSTUNIT.PRG)
CSRC_BAR4	(procedure in INSTUNIT.PRG)
CSRC_BAR5	(procedure in INSTUNIT.PRG)
CSRC_BAR6	(procedure in INSTUNIT.PRG)
CSRC_PRO	(procedure in CREATESR.PRG)
CUST_BAR1	(procedure in INSTUNIT.PRG)
CUST_BAR2	(procedure in INSTUNIT.PRG)
CUST_BAR3	(procedure in INSTUNIT.PRG)
CUST_BAR4	(procedure in INSTUNIT.PRG)
CUST_BAR5	(procedure in INSTUNIT.PRG)
CUST_BAR6	(procedure in INSTUNIT.PRG)
DEF_MENS	(procedure in SURE.PRG)
DELLIN	(procedure in EDITUNIT.PRG)
DELU_BAR1	(procedure in INSTUNIT.PRG)
DELU_BAR2	(procedure in INSTUNIT.PRG)
DELU_BAR3	(procedure in INSTUNIT.PRG)
DELU_BAR4	(procedure in INSTUNIT.PRG)
DISPUNIT	(procedure in CREATESR.PRG)
ECHELON.PRG	
EDITHELP.PRG	
EDIU_BAR1	(procedure in INSTUNIT.PRG)
EDIU_BAR2	(procedure in INSTUNIT.PRG)
EDIU_BAR3	(procedure in INSTUNIT.PRG)
EDIU_BAR4	(procedure in INSTUNIT.PRG)
EDIU_BAR5	(procedure in INSTUNIT.PRG)
EDIU_BAR6	(procedure in INSTUNIT.PRG)
EDIU_BAR7	(procedure in INSTUNIT.PRG)
EDIU_BAR8	(procedure in INSTUNIT.PRG)
EDIU_PRO	(procedure in EDITUNIT.PRG)
ENDBROWS.PRG	
ENDUNIT	(procedure in EDITUNIT.PRG)
HELP.PRG	
HELPNDX.PRG	
HELP_PRO	(procedure in SURE.PRG)
MEMOPAGR()	(function in .PRG)
MEMVARS.PRG	
ORDE_PRO	(procedure in SURE.PRG)
PRNSETUP.PRG	
PROF_PRO	(procedure in SURE.PRG)
QUIT_PRO	(procedure in SURE.PRG)
REINDX.PRG	

EDIT_BAR2	(procedure in INSTTASK.PRG)
EDIT_BAR3	(procedure in INSTTASK.PRG)
EDIT_BAR4	(procedure in INSTTASK.PRG)
EDIT_BAR5	(procedure in INSTTASK.PRG)
EDIT_BAR6	(procedure in INSTTASK.PRG)
EDIT_BAR7	(procedure in INSTTASK.PRG)
EDIT_BAR8	(procedure in INSTTASK.PRG)
DELT_BAR1	(procedure in INSTTASK.PRG)
DELT_BAR2	(procedure in INSTTASK.PRG)
DELT_BAR3	(procedure in INSTTASK.PRG)
STARTASK	(procedure in CREATETA.PRG)
CRET_PRO	(procedure in CREATETA.PRG)
NAMETA	(procedure in CREATETA.PRG)
ASSIGNUN	(procedure in CREATETA.PRG)
ASSIGNU2	(procedure in CREATETA.PRG)
UNITQTY	(procedure in CREATETA.PRG)
ENDTASK	(procedure in CREATETA.PRG)
ENDTASKA	(procedure in CREATETA.PRG)
DISPTASK	(procedure in CREATETA.PRG)
DISPUNI	(procedure in CREATETA.PRG)
COPT_PRO	(procedure in CREATETC.PRG)
FINDTA	(procedure in CREATETC.PRG)
ASSIGNNA	(procedure in CREATETC.PRG)
ENDCOPT	(procedure in CREATETC.PRG)
ENDCOPTA	(procedure in CREATETC.PRG)
EDIT_PRO	(procedure in EDITTASK.PRG)
VIEWTASK	(procedure in EDITTASK.PRG)
VIEWTAS2	(procedure in EDITTASK.PRG)
UNASSIGU	(procedure in EDITTASK.PRG)
UNASSIG2	(procedure in EDITTASK.PRG)
ENDEDIT	(procedure in EDITTASK.PRG)
SETBROW2.PRG	
DELT_PRO	(procedure in DELTASK.PRG)
ENDDELT	(procedure in DELTASK.PRG)
DELTAS	(procedure in DELTASK.PRG)
CREP_BAR1	(procedure in INSTPROF.PRG)
CREP_BAR2	(procedure in INSTPROF.PRG)
CREP_BAR3	(procedure in INSTPROF.PRG)
EDIP_BAR1	(procedure in INSTPROF.PRG)
EDIP_BAR2	(procedure in INSTPROF.PRG)
EDIP_BAR3	(procedure in INSTPROF.PRG)
EDIP_BAR4	(procedure in INSTPROF.PRG)
EDIP_BAR5	(procedure in INSTPROF.PRG)
EDIP_BAR6	(procedure in INSTPROF.PRG)
EDIP_BAR7	(procedure in INSTPROF.PRG)
EDIP_BAR8	(procedure in INSTPROF.PRG)
EDIP_BAR9	(procedure in INSTPROF.PRG)
DELP_BAR1	(procedure in INSTPROF.PRG)
DELP_BAR2	(procedure in INSTPROF.PRG)
DELP_BAR3	(procedure in INSTPROF.PRG)
CRCP_BAR1	(procedure in INSTPROF.PRG)
CRCP_BAR2	(procedure in INSTPROF.PRG)
CRCP_BAR3	(procedure in INSTPROF.PRG)
CRCP_BAR4	(procedure in INSTPROF.PRG)
CRCP_BAR6	(procedure in INSTPROF.PRG)
STARTPRO	(procedure in CREATEPR.PRG)
CREP_PRO	(procedure in CREATEPR.PRG)
NEWPROFI	(procedure in CREATEPR.PRG)
VIEWPROF	(procedure in CREATEPR.PRG)

```
ENDCREP          (procedure in CREATEPR.PRG)
DISPPROF         (procedure in CREATEPR.PRG)
STARTEPR         (procedure in EDITPROF.PRG)
EDIP_PRO         (procedure in EDITPROF.PRG)
SELEPROF         (procedure in EDITPROF.PRG)
CHANGEAP         (procedure in EDITPROF.PRG)
CHANGEAC         (procedure in EDITPROF.PRG)
CHANGEI          (procedure in EDITPROF.PRG)
CHANGEIF         (procedure in EDITPROF.PRG)
CHANGECI         (procedure in EDITPROF.PRG)
CHANGEH2         (procedure in EDITPROF.PRG)
CHANGEME         (procedure in EDITPROF.PRG)
DELP_PRO         (procedure in DELPROF.PRG)
ENDDELP          (procedure in DELPROF.PRG)
DELPRO           (procedure in DELPROF.PRG)
SELECTPRO        (procedure in DELPROF.PRG)
CRCP_PRO         (procedure in CRECUSPR.PRG)
NEWCUSPR         (procedure in CRECUSPR.PRG)
VIEWCUSP         (procedure in CRECUSPR.PRG)
EDITCUSP         (procedure in CRECUSPR.PRG)
ENDCUSPR         (procedure in CRECUSPR.PRG)
DELCUSPR         (procedure in CRECUSPR.PRG)
DISPPRO          (procedure in CRECUSPR.PRG)
ORDS_BAR1        (procedure in INSTORDE.PRG)
ORDS_BAR2        (procedure in INSTORDE.PRG)
ORDS_BAR3        (procedure in INSTORDE.PRG)
ORDS_BAR4        (procedure in INSTORDE.PRG)
ORDS_BAR5        (procedure in INSTORDE.PRG)
TOMP_BAR1        (procedure in INSTORDE.PRG)
TOMP_BAR2        (procedure in INSTORDE.PRG)
TOMP_BAR3        (procedure in INSTORDE.PRG)
TOMP_BAR4        (procedure in INSTORDE.PRG)
TOMP_BAR5        (procedure in INSTORDE.PRG)
TOMP_BAR6        (procedure in INSTORDE.PRG)
ORDT_BAR1        (procedure in INSTORDE.PRG)
ORDT_BAR2        (procedure in INSTORDE.PRG)
ORDT_BAR3        (procedure in INSTORDE.PRG)
ORDT_BAR4        (procedure in INSTORDE.PRG)
ORDT_BAR5        (procedure in INSTORDE.PRG)
ORDT_BAR6        (procedure in INSTORDE.PRG)
ORDT_BAR7        (procedure in INSTORDE.PRG)
ORDT_BAR8        (procedure in INSTORDE.PRG)
DORD_BAR1        (procedure in INSTORDE.PRG)
DORD_BAR2        (procedure in INSTORDE.PRG)
DORD_BAR3        (procedure in INSTORDE.PRG)
STRE_BAR1        (procedure in INSTORDE.PRG)
STRE_BAR2        (procedure in INSTORDE.PRG)
STRE_BAR3        (procedure in INSTORDE.PRG)
STRE_BAR4        (procedure in INSTORDE.PRG)
STRE_BAR5        (procedure in INSTORDE.PRG)
STRE_BAR6        (procedure in INSTORDE.PRG)
STARTOR          (procedure in CREATEOR.PRG)
TOMP_PRO         (procedure in CREATEOR.PRG)
ORDS_PRO         (procedure in CREATEOR.PRG)
CREATEO          (procedure in CREATEOR.PRG)
VIEWORDE         (procedure in CREATEOR.PRG)
ASSIGNTA         (procedure in CREATEOR.PRG)
ENDORDR          (procedure in CREATEOR.PRG)
DISPORD          (procedure in CREATEOR.PRG)
```

```
LISTPROF                (procedure in SYSREP.PRG)
UNITLST                 (procedure in SYSREP.PRG)
UNITLIN                 (procedure in SYSREP.PRG)
PROFLST                 (procedure in SYSREP.PRG)
TASKLST                 (procedure in SYSREP.PRG)
TASKUNI                 (procedure in SYSREP.PRG)
ORDERTP                 (procedure in SYSREP.PRG)
ORDERLST                (procedure in SYSREP.PRG)
ORDERTUP                (procedure in SYSREP.PRG)
RESTORE                 (procedure in RESTOREF.PRG)
GETSTATU                (procedure in RESTOREF.PRG)
PHEL_BAR1               (procedure in INSTPHLP.PRG)
PHEL_BAR2               (procedure in INSTPHLP.PRG)
PHEL_BAR3               (procedure in INSTPHLP.PRG)
PHEL_PRO                (procedure in PRNTHELP.PRG)
PRNONEH                 (procedure in PRNTHELP.PRG)
PRNALLH                 (procedure in PRNTHELP.PRG)
ENDPHEL                 (procedure in PRNTHELP.PRG)
BROWHELP                (procedure in PRNTHELP.PRG)
PRNTMEMO.PRG
STARTPLR                (procedure in PRINTPOL.PRG)
DETA_PRO                (procedure in PRINTPOL.PRG)
POLCALC                 (procedure in PRINTPOL.PRG)
PUNITPOL                (procedure in PRINTPOL.PRG)
PTASKPOL                (procedure in PRINTPOL.PRG)
PORDEPOL                (procedure in PRINTPOL.PRG)
LINPOL                  (procedure in PRINTPOL.PRG)
REPORTHD                (procedure in PRINTPOL.PRG)
TFHEAD                  (procedure in PRINTPOL.PRG)
DETA_BAR1               (procedure in PRINTPOL.PRG)
DETA_BAR2               (procedure in PRINTPOL.PRG)
DETA_BAR3               (procedure in PRINTPOL.PRG)
DETA_BAR4               (procedure in PRINTPOL.PRG)
PRIN_BAR1               (procedure in PRINTPOL.PRG)
PRIN_BAR2               (procedure in PRINTPOL.PRG)
PRIN_BAR3               (procedure in PRINTPOL.PRG)
PRIN_BAR4               (procedure in PRINTPOL.PRG)
PRIN_BAR5               (procedure in PRINTPOL.PRG)
AMMOCALC                (procedure in PRINTAMM.PRG)
PUNITAMM                (procedure in PRINTAMM.PRG)
PTASKAMM                (procedure in PRINTAMM.PRG)
PORDEAMM                (procedure in PRINTAMM.PRG)
LINAMM                  (procedure in PRINTAMM.PRG)
OTHECALC                (procedure in PRINTOTH.PRG)
PUNITOTH                (procedure in PRINTOTH.PRG)
PTASKOTH                (procedure in PRINTOTH.PRG)
PORDEOTH                (procedure in PRINTOTH.PRG)
OTHERCAL                (procedure in CALCOTHE.PRG)

Procedure files:
  INSTRUCT.PRG
  SURE.PRG
  INSTUNIT.PRG
  CREATESR.PRG
  EDITUNIT.PRG
  CUSTUNIT.PRG
  DELUNIT.PRG
  INSTTASK.PRG
  CREATETA.PRG
```

```
CREATETC.PRG
EDITTASK.PRG
DELTASK.PRG
INSTPROF.PRG
CREATEPR.PRG
EDITPROF.PRG
DELPROF.PRG
CRECUSPR.PRG
INSTORDE.PRG
CREATEOR.PRG
EDITORDE.PRG
DELORDE.PRG
INSTREPT.PRG
ORDEREP1.PRG
INSTSYSR.PRG
SYSREP.PRG
RESTOREF.PRG
INSTPHLP.PRG
PRNTHELP.PRG
PRINTPOL.PRG
PRINTAMM.PRG
PRINTOTH.PRG
CALCPOL.PRG
CALCAMMO.PRG
CALCOTHE.PRG
```

Databases:
```
&FILEVAR.DBF
&FILEVAR2.DBF
AMMOEST.DBF
AMMORATE.DBF
CLASS_I.DBF
CL_OTHER.DBF
CPROFILE.DBF
CSRCLINQ.DBF
CSRCLIST.DBF
DODIC.DBF
ECHELONS.DBF
HELP.DBF
HELP.DBT
INTENSTY.DBF
LINLIST.DBF
MEDICAL.DBF
ORDERLST.DBF
ORDERTAS.DBF
POLEST.DBF
POLRATE.DBF
POLWGT.DBF
POSTURE.DBF
PROFILES.DBF
SRCLINQT.DBF
SRCLIST.DBF
TASKORG.DBF
TEMP.DBF
TTEMP.DBF
TTTEMP.DBF
WATER.DBF
```

Multiple index files:

```
&FILEVAR
AMMOEST.MDX
AMMORATE.MDX
CLASS_I.MDX
CL_OTHER.MDX
CPROFILE.MDX
CSRCLINQ.MDX
CSRCLIST.MDX
DODIC.MDX
ECHELONS.MDX
HELP.MDX
INTENSTY.MDX
LINLIST.MDX
MEDICAL.MDX
ORDERLST.MDX
ORDERTAS.MDX
POLEST.MDX
POLRATE.MDX
POLWGT.MDX
PROFILES.MDX
SRCLINQT.MDX
SRCLIST.MDX
TASKORG.MDX
WATER.MDX

Memory files:
VARIABLE.MEM
```

---

29 databases in the system

    HELP.DBF
    CSRCLIST.DBF
    TEMP.DBF
    SRCLIST.DBF
    SRCLINQT.DBF
    CSRCLINQ.DBF
    TTEMP.DBF
    TTTEMP.DBF
    ECHELONS.DBF
    LINLIST.DBF
    AMMORATE.DBF
    POLRATE.DBF
    TASKORG.DBF
    ORDERTAS.DBF
    MEDICAL.DBF
    WATER.DBF
    CLASS_I.DBF
    POSTURE.DBF
    INTENSTY.DBF
    PROFILES.DBF
    CPROFILE.DBF
    ORDERLST.DBF
    CL_OTHER.DBF
    POLWGT.DBF
    DODIC.DBF
    POLEST.DBF
    AMMOEST.DBF
    &FILEVAR.DBF
    &FILEVAR2.DBF

Structure for database : HELP.DBF
Number of data records :   28
        Last updated : 04/10/94 at 13:49

Field	Field name	Type	Width	Dec	Start	End
1	PROGRAM	Character	8		1	8
2	NOTES	Memo	10		9	18
3	SUBJECT	Character	60		19	78
4	MENU	Memo	10		79	88
** Total **			89			

This database is associated with the memo file: HELP.DBT

This database is associated with index file/tag(s):
        : PROGRAM   (is an index tag)
        : SUBJECT   (is an index tag)

This database is associated with multiple index file(s):
        : HELP.MDX

Used by: STARTHLP.PRG
        : REINDX.PRG

379

```
-------------------------------------------------------------------

      Structure for database : CSRCLIST.DBF
      Number of data records :        3
              Last updated : 04/16/94 at 10:49
      Field  Field name  Type        Width   Dec    Start      End
         1   SRC         Character      9               1        9
         2   UNIT_DESIG  Character     38              10       47
         3   STRENGTH    Numeric        6              48       53
         4   UNITNAME    Character     60              54      113
         5   TASKORG     Logical        1             114      114
         6   SUBORDINAT  Logical        1             115      115
         7   TASKORGC    Logical        1             116      116
         8   ECHELON     Character      9             117      125
         9   QTY         Numeric        2             126      127
      ** Total **                     128


      This database is associated with index file/tag(s):
              : SRC   (is an index tag)
              : UNIT_DESIG  (is an index tag)
              : STRENGTH  (is an index tag)
              : UNITNAME  (is an index tag)
              : ECHELON  (is an index tag)

      This database is associated with multiple index file(s):
              : CSRCLIST.MDX

      Used by: REINDX.PRG
              : BACKDBF.PRG
              : STARTCSR            (procedure in CREATESR.PRG)
              : STARTUNI            (procedure in EDITUNIT.PRG)
              : STARTED             (procedure in DELUNIT.PRG)
              : STARTASK            (procedure in CREATETA.PRG)
              : CALCPOL.PRG
              : CALCAMMO.PRG
              : CALCOTHE.PRG
              : STARTSRP            (procedure in SYSREP.PRG)
              : STARTPLR            (procedure in PRINTPOL.PRG)


-------------------------------------------------------------------

      Structure for database : TEMP.DBF
      Number of data records :        0
              Last updated : 04/16/94 at 10:57
      Field  Field name  Type        Width   Dec    Start      End
         1   SRC         Character      9               1        9
         2   UNIT_DESIG  Character     38              10       47
         3   STRENGTH    Numeric        6              48       53
         4   UNITNAME    Character     60              54      113
         5   TASKORG     Logical        1             114      114
         6   SUBORDINAT  Logical        1             115      115
         7   TASKORGC    Logical        1             116      116
         8   ECHELON     Character      9             117      125
         9   QTY         Numeric        2             126      127
      ** Total **                     128
```

NO associated index files

NO associated multiple indexes

Used by: STARTCSR   (procedure in CREATESR.PRG)
       : UNITDATA   (procedure in CUSTUNIT.PRG)
       : ASSIGNUN   (procedure in CREATETA.PRG)
       : ASSIGNU2   (procedure in CREATETA.PRG)
       : ENDTASK   (procedure in CREATETA.PRG)
       : ENDTASKA   (procedure in CREATETA.PRG)
       : ENDCOPT   (procedure in CREATETC.PRG)
       : ENDCOPTA   (procedure in CREATETC.PRG)
       : NEWPROFI   (procedure in CREATEPR.PRG)
       : NEWCUSPR   (procedure in CRECUSPR.PRG)
       : CREATEO   (procedure in CREATEOR.PRG)
       : ASSIGNTA   (procedure in CREATEOR.PRG)

Structure for database : SRCLIST.DBF
Number of data records :   997
         Last updated : 03/31/94 at 15:04

Field	Field name	Type	Width	Dec	Start	End
1	SRC	Character	9		1	9
2	UNIT_DESIG	Character	38		10	47
3	STRENGTH	Numeric	6		48	53
4	HIDE	Logical	1		54	54
5	QTY	Numeric	2		55	56
** Total **			57			

This database is associated with index file/tag(s):
       : SRC   (is an index tag)
       : UNIT_DESIG   (is an index tag)

This database is associated with multiple index file(s):
       : SRCLIST.MDX

Used by: REINDX.PRG
       : STARTCSR   (procedure in CREATESR.PRG)
       : STARTUNI   (procedure in EDITUNIT.PRG)
       : STARTSRP   (procedure in SYSREP.PRG)

Structure for database : SRCLINQT.DBF
Number of data records :   21622
         Last updated : 03/31/94 at 15:01

Field	Field name	Type	Width	Dec	Start	End
1	SRC	Character	9		1	9
2	LIN	Character	6		10	15
3	QUANTITY	Numeric	6		16	21
** Total **			22			

381

```
This database is associated with index file/tag(s):
        : SRC   (is an index tag)
        : LIN   (is an index tag)
        : UNIQUELI   (is an index tag)

This database is associated with multiple index file(s):
        : SRCLINQT.MDX

Used by: REINDX.PRG
        : STARTCSR              (procedure in CREATESR.PRG)
        : STARTUNI              (procedure in EDITUNIT.PRG)
        : STARTSRP              (procedure in SYSREP.PRG)

------------------------------------------------------------------

Structure for database : CSRCLINQ.DBF
Number of data records :      108
        Last updated : 04/16/94 at 10:49
Field  Field name  Type        Width   Dec    Start      End
   1   SRC         Character      9              1         9
   2   LIN         Character      6             10        15
   3   QUANTITY    Numeric        6             16        21
   4   UNITNAME    Character     60             22        81
** Total **                     82

This database is associated with index file/tag(s):
        : SRC   (is an index tag)
        : LIN   (is an index tag)
        : UNITNAME   (is an index tag)

This database is associated with multiple index file(s):
        : CSRCLINQ.MDX

Used by: REINDX.PRG
        : BACKDBF.PRG
        : STARTCSR              (procedure in CREATESR.PRG)
        : STARTUNI              (procedure in EDITUNIT.PRG)
        : STARTED               (procedure in DELUNIT.PRG)
        : CALCPOL.PRG
        : CALCAMMO.PRG
        : CALCOTHE.PRG
        : STARTSRP              (procedure in SYSREP.PRG)
        : STARTPLR              (procedure in PRINTPOL.PRG)

------------------------------------------------------------------

Structure for database : TTEMP.DBF
Number of data records :       21
        Last updated : 04/16/94 at 10:49
Field  Field name  Type        Width   Dec    Start      End
   1   SRC         Character      9              1         9
   2   LIN         Character      6             10        15
   3   QUANTITY    Numeric        6             16        21
   4   UNITNAME    Character     60             22        81
** Total **                     82
```

NO associated index files

NO associated multiple indexes

Used by: COPYLINS   (procedure in CREATESR.PRG)
       : ASSIGNUN   (procedure in CREATETA.PRG)
       : ASSIGNU2   (procedure in CREATETA.PRG)
       : ENDTASK   (procedure in CREATETA.PRG)
       : ENDTASKA   (procedure in CREATETA.PRG)
       : ASSIGNTA   (procedure in CREATEOR.PRG)

Structure for database : TTTEMP.DBF
Number of data records :   21
        Last updated : 04/16/94 at 10:49
Field Field name Type   Width   Dec   Start   End
    1 SRC   Character   9   1   9
    2 LIN   Character   6   10   15
    3 QUANTITY   Numeric   6   16   21
** Total **   22

NO associated index files

NO associated multiple indexes

Used by: COPYLINS   (procedure in CREATESR.PRG)

Structure for database : ECHELONS.DBF
Number of data records :   4
        Last updated : 01/28/94 at 21:23
Field Field name Type   Width   Dec   Start   End
    1 ECHELON   Character   21   1   21
** Total **   22

This database is associated with index file/tag(s):
       : ECHELON   (is an index tag)

This database is associated with multiple index file(s):
       : ECHELONS.MDX

Used by: REINDX.PRG
       : ADDDATA   (procedure in CREATESR.PRG)
       : STARTUNI   (procedure in EDITUNIT.PRG)
       : CHANGEEH   (procedure in EDITUNIT.PRG)
       : CALCOTHE.PRG

Structure for database : LINLIST.DBF
Number of data records :   471

383

```
              Last updated : 03/25/94 at   8:35
Field   Field name   Type        Width   Dec   Start      End
    1   LIN          Character      6             1         6
    2   NOMEN        Character     64             7        70
    3   AMMO_RATE    Logical        1            71        71
    4   POL_RATE     Logical        1            72        72
    5   QTY          Numeric        4            73        76
** Total **                        77
```

This database is associated with index file/tag(s):
        : LIN   (is an index tag)

This database is associated with multiple index file(s):
        : LINLIST.MDX

Used by: REINDX.PRG
        : STARTUNI   (procedure in EDITUNIT.PRG)

```
Structure for database : AMMORATE.DBF
Number of data records :        608
            Last updated : 03/31/94 at 14:56
Field   Field name   Type        Width   Dec   Start      End
    1   LIN          Character      9             1         9
    2   LIN_NOMEN    Character     39            10        48
    3   DODIC        Character      8            49        56
    4   DODIC_NOM    Character     29            57        85
    5   HVY1D        Numeric        9     2       86        94
    6   HVYSD        Numeric        9     2       95       103
    7   HVY1A        Numeric        9     2      104       112
    8   HVYSA        Numeric        9     2      113       121
    9   HVYP         Numeric        9     2      122       130
   10   WEIGHT       Numeric       10     2      131       140
   11   CUBE         Numeric        8     3      141       148
   12   CBT_LOAD     Numeric        8           149       156
** Total **                       157
```

This database is associated with index file/tag(s):
        : LIN   (is an index tag)
        : DODIC   (is an index tag)

This database is associated with multiple index file(s):
        : AMMORATE.MDX

Used by: REINDX.PRG
        : STARTUNI   (procedure in EDITUNIT.PRG)
        : CALCAMMO.PRG
        : STARTSRP   (procedure in SYSREP.PRG)
        : STARTPLR   (procedure in PRINTPOL.PRG)

Structure for database : POLRATE.DBF
Number of data records :   467

384

```
              Last updated : 03/31/94 at 14:57
Field   Field name  Type        Width   Dec     Start       End
    1   IDLE_AVG    Numeric       12      3         1         12
    2   CC          Numeric        8      3        13         20
    3   SR          Numeric        8      3        21         28
    4   FT_CAP      Numeric        9      2        29         37
    5   LIN         Character      9               38         46
    6   LIN_NOMEN   Character     39               47         85
    7   EQUIP_CAT   Character      4               86         89
    8   POL_TYPE    Character      3               90         92
** Total **                      93
```

This database is associated with index file/tag(s):
        : LIN   (is an index tag)
        : POL_TYPE   (is an index tag)

This database is associated with multiple index file(s):
        : POLRATE.MDX

Used by: REINDX.PRG
        : STARTUNI   (procedure in EDITUNIT.PRG)
        : CALCPOL.PRG
        : STARTSRP   (procedure in SYSREP.PRG)
        : STARTPLR   (procedure in PRINTPOL.PRG)

```
Structure for database : TASKORG.DBF
Number of data records :        0
              Last updated : 04/15/94 at 19:47
Field   Field name  Type        Width   Dec     Start       End
    1   TASKORGNAM  Character     60                1         60
    2   UNITNAME    Character     60               61        120
    3   CHANGESTAT  Logical        1              121        121
    4   CHANGESTA2  Logical        1              122        122
    5   QTY         Numeric        2              123        124
    6   SRC         Character      9              125        133
** Total **                     134
```

This database is associated with index file/tag(s):
        : TASKORGNAM (is an index tag)
        : UNITNAME   (is an index tag)
        : UNIQUEUN   (is an index tag)
        : UNIQUETA   (is an index tag)
        : SRC   (is an index tag)

This database is associated with multiple index file(s):
        : TASKORG.MDX

Used by: REINDX.PRG
        : BACKDBF.PRG
        : STARTED   (procedure in DELUNIT.PRG)
        : STARTASK   (procedure in CREATETA.PRG)
        : STARTOR   (procedure in CREATEOR.PRG)
        : CALCPOL.PRG
        : CALCAMMO.PRG

                        385
```

```
        :  CALCOTHE.PRG
        :  STARTSRP                    (procedure in SYSREP.PRG)
        :  STARTPLR                    (procedure in PRINTPOL.PRG)


------------------------------------------------------------------


Structure for database : ORDERTAS.DBF
Number of data records :         0
          Last updated : 04/15/94 at 19:46
Field  Field name  Type        Width   Dec    Start      End
   1   NUMBER      Character       8            1          8
   2   TASKORGNAM  Character      60            9         68
   3   PHASE       Numeric         1           69         69
   4   PROFILENAM  Character      20           70         89
   5   STRENGTH    Numeric         3           90         92
   6   CHANGESTAT  Logical         1           93         93
** Total **                      94


This database is associated with index file/tag(s):
        : NUMBER   (is an index tag)
        : TASKORGNAM  (is an index tag)
        : TASKPHAS  (is an index tag)
        : PROFILENAM  (is an index tag)
        : TASKPHASE  (is an index tag)
        : UNIQUENU  (is an index tag)
        : UNIQUETA  (is an index tag)
        : PHASE  (is an index tag)
        : UNIQUETN  (is an index tag)
        : PHASETAS  (is an index tag)

This database is associated with multiple index file(s):
        : ORDERTAS.MDX

Used by: DELORDE.PRG
        : REINDX.PRG
        : BACKDBF.PRG
        : STARTED                     (procedure in DELUNIT.PRG)
        : STARTOR                     (procedure in CREATEOR.PRG)
        : CALCPOL.PRG
        : CALCAMMO.PRG
        : CALCOTHE.PRG
        : STARTSRP                    (procedure in SYSREP.PRG)
        : STARTPLR                    (procedure in PRINTPOL.PRG)


------------------------------------------------------------------


Structure for database : MEDICAL.DBF
Number of data records :        12
          Last updated : 03/08/94 at 16:35
Field  Field name  Type        Width   Dec    Start      End
   1   ECHELON     Character      14            1         14
   2   INTENSITY   Character       9           15         23
   3   PLAN_FACT   Numeric         4     2     24         27
   4   MED_NAME    Character      25           28         52
** Total **                      53
```

This database is associated with index file/tag(s):
        : MED_NAME (is an index tag)
        : ECHELON (is an index tag)

This database is associated with multiple index file(s):
        : MEDICAL.MDX

Used by: REINDX.PRG
        : STARTPRO   (procedure in CREATEPR.PRG)
        : STARTEPR   (procedure in EDITPROF.PRG)
        : CALCOTHE.PRG
        : STARTPLR   (procedure in PRINTPOL.PRG)

Structure for database : WATER.DBF
Number of data records :   20
        Last updated : 03/08/94 at 16:35
Field Field name Type   Width   Dec   Start   End
    1 ECHELON   Character   9   1   9
    2 CLIMATE   Character   9   10   18
    3 GAL_MAN_DA Numeric   4   1   19   22
    4 WATER_NAME Character   19   23   41
** Total **   42

This database is associated with index file/tag(s):
        : ECHELON (is an index tag)
        : WATER_NAME (is an index tag)

This database is associated with multiple index file(s):
        : WATER.MDX

Used by: REINDX.PRG
        : STARTPRO   (procedure in CREATEPR.PRG)
        : STARTEPR   (procedure in EDITPROF.PRG)
        : CALCOTHE.PRG
        : STARTPLR   (procedure in PRINTPOL.PRG)

Structure for database : CLASS_I.DBF
Number of data records :   6
        Last updated : 04/10/94 at 19:06
Field Field name Type   Width   Dec   Start   End
    1 RATION   Character   10   1   10
    2 PLAN_FACT   Numeric   5   3   11   15
    3 UNIT   Character   12   16   27
    4 UNIT_QTY   Numeric   9   28   36
    5 TASK_ORG_Q Numeric   9   37   45
    6 ORDER_QTY   Numeric   10   46   55
    7 ORDER_TTL   Numeric   10   56   65
** Total **   66

This database is associated with index file/tag(s):

,   : RATION   (is an index tag)

This database is associated with multiple index file(s):
        : CLASS_I.MDX

Used by: REINDX.PRG
        : STARTPRO   (procedure in CREATEPR.PRG)
        : STARTEPR   (procedure in EDITPROF.PRG)
        : STARTPLR   (procedure in PRINTPOL.PRG)

Structure for database : POSTURE.DBF
Number of data records :   6
        Last updated : 03/08/94 at 16:35
Field   Field name   Type   Width   Dec   Start   End
    1   POSTURE   Character   25   1   25
** Total **   26

NO associated index files

NO associated multiple indexes

Used by: STARTPRO   (procedure in CREATEPR.PRG)
        : STARTEPR   (procedure in EDITPROF.PRG)
        : CALCPOL.PRG
        : CALCAMMO.PRG
        : CALCOTHE.PRG
        : STARTPLR   (procedure in PRINTPOL.PRG)

Structure for database : INTENSTY.DBF
Number of data records :   5
        Last updated : 03/30/94 at 7:30
Field   Field name   Type   Width   Dec   Start   End
    1   INTENSITY   Character   8   1   8
    2   FACTOR   Numeric   5   2   9   13
** Total **   14

NO associated index files

This database is associated with multiple index file(s):
        : INTENSTY.MDX

Used by: BACKDBF.PRG
        : STARTPRO   (procedure in CREATEPR.PRG)
        : STARTEPR   (procedure in EDITPROF.PRG)
        : CALCPOL.PRG
        : CALCAMMO.PRG
        : CALCOTHE.PRG
        : STARTPLR   (procedure in PRINTPOL.PRG)

388

```
Structure for database : PROFILES.DBF
Number of data records :        12
         Last updated : 03/29/94 at   9:14
   Field  Field name  Type        Width   Dec    Start      End
      1    PROFILE     Character      15             1        15
      2    AE_HRS      Numeric        5     1        16        20
      3    AV_HRS      Numeric        5     1        21        25
      4    CE_HRS      Numeric        5     1        26        30
      5    GN_HRS      Numeric        5     1        31        35
      6    HG_HRS      Numeric        5     1        36        40
      7    MH_HRS      Numeric        5     1        41        45
      8    OV_HRS      Numeric        5     1        46        50
      9    SG_HRS      Numeric        5     1        51        55
     10    SV_HRS      Numeric        5     1        56        60
     11    WV_KMS      Numeric        5     1        61        65
     12    TV_TI_HRS   Numeric        5     1        66        70
     13    TV_CC_HRS   Numeric        5     1        71        75
     14    TV_SR_HRS   Numeric        5     1        76        80
     15    CUSTOM      Logical        1              81        81
** Total **                         82
```

This database is associated with index file/tag(s):
      : PROFILE   (is an index tag)

This database is associated with multiple index file(s):
      : PROFILES.MDX

```
Used by: REINDX.PRG
       : BACKDBF.PRG
       : STARTPRO              (procedure in CREATEPR.PRG)
       : STARTEPR              (procedure in EDITPROF.PRG)
       : CALCPOL.PRG
       : CALCAMMO.PRG
       : CALCOTHE.PRG
       : STARTPLR              (procedure in PRINTPOL.PRG)
```

```
Structure for database : CPROFILE.DBF
Number of data records :         0
         Last updated : 04/15/94 at 19:50
   Field  Field name  Type        Width   Dec    Start      End
      1    PROFILE     Character      15             1        15
      2    INTENSITY   Character       8            16        23
      3    FACTOR      Numeric         5     2      24        28
      4    PROFILENAM  Character      20            29        48
      5    PROFILEDES  Character      50            49        98
      6    POSTURE     Character      25            99       123
      7    CL_I_1      Character       8           124       131
      8    CL_I_2      Character       8           132       139
      9    CL_I_3      Character       8           140       147
     10    CL_I_H1     Logical         1           148       148
     11    CL_I_H2     Logical         1           149       149
     12    MED_NAME    Character      25           150       174
     13    WATER_NAME  Character      19           175       193
     14    CDE         Character       4           194       197
```

This database is associated with index file/tag(s):
       : PROFILE   (is an index tag)
       : INTENSITY   (is an index tag)
       : PROFILENAM   (is an index tag)
       : PROFILEDES   (is an index tag)
       : POSTURE   (is an index tag)
       : MED_NAME   (is an index tag)
       : WATER_NAME   (is an index tag)
       : CDE   (is an index tag)

This database is associated with multiple index file(s):
       : CPROFILE.MDX

Used by: REINDX.PRG
       : BACKDBF.PRG
       : STARTED   (procedure in DELUNIT.PRG)
       : STARTPRO   (procedure in CREATEPR.PRG)
       : STARTEPR   (procedure in EDITPROF.PRG)
       : STARTOR   (procedure in CREATEOR.PRG)
       : CALCPOL.PRG
       : CALCAMMO.PRG
       : CALCOTHE.PRG
       : STARTSRP   (procedure in SYSREP.PRG)
       : STARTPLR   (procedure in PRINTPOL.PRG)

Structure for database : ORDERLST.DBF
Number of data records :   0
        Last updated : 04/15/94 at 19:47

Field	Field name	Type	Width	Dec	Start	End
1	NUMBER	Character	8		1	8
2	NAME	Character	50		9	58
3	DESCRIPTIO	Character	60		59	118
4	PHASES	Numeric	1		119	119
5	PHASE1LEN	Numeric	5	2	120	124
6	PHASE2LEN	Numeric	5	2	125	129
7	PHASE3LEN	Numeric	5	2	130	134
8	PHASE4LEN	Numeric	5	2	135	139
9	PHASE5LEN	Numeric	5	2	140	144
10	PHASE6LEN	Numeric	5	2	145	149
11	PHASE7LEN	Numeric	5	2	150	154
12	PHASE8LEN	Numeric	5	2	155	159
13	PHASE9LEN	Numeric	5	2	160	164

** Total **   165

This database is associated with index file/tag(s):
       : NUMBER   (is an index tag)
       : NAME   (is an index tag)
       : DESCRIPTIO   (is an index tag)

This database is associated with multiple index file(s):
       : ORDERLST.MDX

```
Used by: DELORDE.PRG
       : REINDX.PRG
       : BACKDBF.PRG
       : STARTOR            (procedure in CREATEOR.PRG)
       : STARTORD           (procedure in ORDEREP1.PRG)
       : CALCPOL.PRG
       : CALCAMMO.PRG
       : CALCOTHE.PRG
       : STARTSRP           (procedure in SYSREP.PRG)
       : STARTPLR           (procedure in PRINTPOL.PRG)
```

```
Structure for database : CL_OTHER.DBF
Number of data records :      2
         Last updated : 04/12/94 at 21:59
```

Field	Field name	Type	Width	Dec	Start	End
1	NUMBER	Character	8		1	8
2	UNITNAME	Character	60		9	68
3	TASKORGNAM	Character	60		69	128
4	PHASE	Numeric	1		129	129
5	PROFILENAM	Character	20		130	149
6	CL_I_QTY	Numeric	9		150	158
7	H2O_QTY	Numeric	9		159	167
8	CL_IVBQTY	Numeric	9		168	176
9	CL_IVCQTY	Numeric	9		177	185
10	CL_VIII_QT	Numeric	9		186	194
11	QTY	Numeric	2		195	196
12	SRC	Character	9		197	205
13	STRENGTH	Numeric	6		206	211
** Total **			212			

```
This database is associated with index file/tag(s):
       : UNITNAME   (is an index tag)
       : TASKORGNAM   (is an index tag)
       : PROFILENAM   (is an index tag)

This database is associated with multiple index file(s):
       : CL_OTHER.MDX

Used by: REINDX.PRG
       : STARTORD            (procedure in ORDEREP1.PRG)
       : CALCOTHE.PRG
       : STARTPLR            (procedure in PRINTPOL.PRG)
```

```
Structure for database : POLWGT.DBF
Number of data records :      3
         Last updated : 02/27/94 at 12:39
```

Field	Field name	Type	Width	Dec	Start	End
1	POL_TYPE	Character	3		1	3
2	LBS_GAL	Numeric	9	3	4	12
3	CALC_QTY	Numeric	7		13	19
** Total **			20			

This database is associated with index file/tag(s):
        : POL_TYPE (is an index tag)

This database is associated with multiple index file(s):
        : POLWGT.MDX

Used by: REINDX.PRG
        : CALCPOL.PRG
        : STARTPLR   (procedure in PRINTPOL.PRG)

Structure for database : DODIC.DBF
Number of data records :   184
        Last updated : 04/10/94 at 19:37

Field	Field name	Type	Width	Dec	Start	End
1	DODIC	Character	8		1	8
2	DODIC_NOM	Character	29		9	37
3	WEIGHT	Numeric	10	2	38	47
4	CUBE	Numeric	8	3	48	55
5	CALC_QTY	Numeric	10		56	65
6	CBT_LOADQT	Numeric	9		66	74
7	TASKORG_QT	Numeric	11		75	85
8	ORDER_QT	Numeric	12		86	97
9	ORDER_QTTL	Numeric	12		98	109
10	TASKORG_CB	Numeric	11		110	120
11	ORDER_CB	Numeric	12		121	132
12	ORDER_CBTL	Numeric	12		133	144
13	UNIT_QT	Numeric	11		145	155
14	UNIT_CB	Numeric	11		156	166
** Total **			167			

This database is associated with index file/tag(s):
        : DODIC (is an index tag)

This database is associated with multiple index file(s):
        : DODIC.MDX

Used by: REINDX.PRG
        : CALCAMMO.PRG
        : STARTPLR   (procedure in PRINTPOL.PRG)

Structure for database : POLEST.DBF
Number of data records :   139
        Last updated : 04/12/94 at 21:58

Field	Field name	Type	Width	Dec	Start	End
1	NUMBER	Character	8		1	8
2	UNITNAME	Character	60		9	68
3	TASKORGNAM	Character	60		69	128
4	PHASE	Numeric	1		129	129
5	PARENT_UNI	Character	60		130	189
6	LIN	Character	6		190	195
7	QUANTITY	Numeric	6		196	201

392

	8	POL_TYPE	Character	3		202	204
	9	POL_QTY	Numeric	9	2	205	213
	10	LIN_NOMEN	Character	39		214	252
	11	FT_CAP	Numeric	11	2	253	263
	12	DONE	Logical	1		264	264
	13	QTY	Numeric	2		265	266
	14	SRC	Character	9		267	275
	15	EQUIP_CAT	Character	4		276	279
	16	LBS_GAL	Numeric	9	3	280	288
** Total **				289			

This database is associated with index file/tag(s):
: NUMBER (is an index tag)
: UNITNAME (is an index tag)
: TASKORGNAM (is an index tag)
: LIN (is an index tag)
: POL_TYPE (is an index tag)
: PARENT_UNI (is an index tag)

This database is associated with multiple index file(s):
: POLEST.MDX

Used by: REINDX.PRG
: CALCPOL.PRG
: STARTPLR   (procedure in PRINTPOL.PRG)

Structure for database : AMMOEST.DBF
Number of data records :   148
        Last updated : 04/12/94 at 21:59

Field	Field name	Type	Width	Dec	Start	End
1	NUMBER	Character	8		1	8
2	UNITNAME	Character	60		9	68
3	TASKORGNAM	Character	60		69	128
4	PHASE	Numeric	1		129	129
5	LIN	Character	6		130	135
6	QUANTITY	Numeric	6		136	141
7	DODIC	Character	8		142	149
8	CALC_QTY	Numeric	12	2	150	161
9	CBT_LOADQT	Numeric	9		162	170
10	LIN_NOMEN	Character	39		171	209
11	DODIC_NOM	Character	29		210	238
12	QTY	Numeric	2		239	240
13	SRC	Character	9		241	249
** Total **			250			

This database is associated with index file/tag(s):
: NUMBER (is an index tag)
: UNITNAME (is an index tag)
: TASKORGNAM (is an index tag)
: DODIC (is an index tag)
: LIN (is an index tag)
: UNITDODI (is an index tag)

This database is associated with multiple index file(s):

```
            :  AMMOEST.MDX

Used by:  REINDX.PRG
          :  CALCAMMO.PRG
          :  STARTPLR                      (procedure in PRINTPOL.PRG)

    --------------------------------------------------------------


&FILEVAR.DBF is a macro unknown to SNAP!


NO associated index files

NO associated multiple indexes

Used by:  BACKDBF.PRG
          :  RESTORE                       (procedure in RESTOREF.PRG)

    --------------------------------------------------------------


&FILEVAR2.DBF is a macro

NO associated index files

NO associated multiple indexes

Used by:  RESTORE                          (procedure in RESTOREF.PRG)
```

Field Name	Type	Len	Dec	Database
AE_HRS	N	5	1	PROFILES.DBF
AMMO_RATE	L	1	0	LINLIST.DBF
AV_HRS	N	5	1	PROFILES.DBF
CALC_QTY	N	12	2	AMMOEST.DBF
CALC_QTY	N	10	0	DODIC.DBF
CALC_QTY	N	7	0	POLWGT.DBF
CBT_LOAD	N	8	0	AMMORATE.DBF
CBT_LOADQT	N	9	0	DODIC.DBF
				AMMOEST.DBF
CC	N	8	3	POLRATE.DBF
CDE	C	4	0	CPROFILE.DBF
CE_HRS	N	5	1	PROFILES.DBF
CHANGESTA2	L	1	0	TASKORG.DBF
CHANGESTAT	L	1	0	ORDERTAS.DBF
				TASKORG.DBF
CLIMATE	C	9	0	WATER.DBF
CL_IVBQTY	N	9	0	CL_OTHER.DBF
CL_IVCQTY	N	9	0	CL_OTHER.DBF
CL_I_1	C	8	0	CPROFILE.DBF
CL_I_2	C	8	0	CPROFILE.DBF
CL_I_3	C	8	0	CPROFILE.DBF
CL_I_H1	L	1	0	CPROFILE.DBF
CL_I_H2	L	1	0	CPROFILE.DBF
CL_I_QTY	N	9	0	CL_OTHER.DBF
CL_VIII_QT	N	9	0	CL_OTHER.DBF
CUBE	N	8	3	DODIC.DBF
				AMMORATE.DBF
CUSTOM	L	1	0	PROFILES.DBF
DESCRIPTIO	C	60	0	ORDERLST.DBF
DODIC	C	8	0	DODIC.DBF
				AMMOEST.DBF
				AMMORATE.DBF
DODIC_NOM	C	29	0	AMMOEST.DBF
				DODIC.DBF
				AMMORATE.DBF
DONE	L	1	0	POLEST.DBF
ECHELON	C	9	0	WATER.DBF
ECHELON	C	14	0	MEDICAL.DBF
ECHELON	C	9	0	CSRCLIST.DBF
ECHELON	C	21	0	ECHELONS.DBF
ECHELON	C	9	0	TEMP.DBF
EQUIP_CAT	C	4	0	POLRATE.DBF
				POLEST.DBF
FACTOR	N	5	2	INTENSTY.DBF
				CPROFILE.DBF
FT_CAP	N	11	2	POLEST.DBF
FT_CAP	N	9	2	POLRATE.DBF
GAL_MAN_DA	N	4	1	WATER.DBF
GN_HRS	N	5	1	PROFILES.DBF
H2O_QTY	N	9	0	CL_OTHER.DBF
HG_HRS	N	5	1	PROFILES.DBF
HIDE	L	1	0	SRCLIST.DBF

HVY1A	N	9	2	AMMORATE.DBF
HVY1D	N	9	2	AMMORATE.DBF
HVYP	N	9	2	AMMORATE.DBF
HVYSA	N	9	2	AMMORATE.DBF
HVYSD	N	9	2	AMMORATE.DBF
IDLE_AVG	N	12	3	POLRATE.DBF
INTENSITY	C	9	0	MEDICAL.DBF
INTENSITY	C	8	0	INTENSTY.DBF
				CPROFILE.DBF
LBS_GAL	N	9	3	POLWGT.DBF
				POLEST.DBF
LIN	C	6	0	LINLIST.DBF
				AMMOEST.DBF
				CSRCLINQ.DBF
				TTTEMP.DBF
LIN	C	9	0	AMMORATE.DBF
LIN	C	6	0	TTEMP.DBF
LIN	C	9	0	POLRATE.DBF
LIN	C	6	0	POLEST.DBF
				SRCLINQT.DBF
LIN_NOMEN	C	39	0	POLEST.DBF
				POLRATE.DBF
				AMMORATE.DBF
				AMMOEST.DBF
MED_NAME	C	25	0	CPROFILE.DBF
				MEDICAL.DBF
MENU	M	10	0	HELP.DBF
MH_HRS	N	5	1	PROFILES.DBF
NAME	C	50	0	ORDERLST.DBF
NOMEN	C	64	0	LINLIST.DBF
NOTES	M	10	0	HELP.DBF
NUMBER	C	8	0	POLEST.DBF
				AMMOEST.DBF
				CL_OTHER.DBF
				ORDERLST.DBF
				ORDERTAS.DBF
ORDER_CB	N	12	0	DODIC.DBF
ORDER_CBTL	N	12	0	DODIC.DBF
ORDER_QT	N	12	0	DODIC.DBF
ORDER_QTTL	N	12	0	DODIC.DBF
ORDER_QTY	N	10	0	CLASS_I.DBF
ORDER_TTL	N	10	0	CLASS_I.DBF
OV_HRS	N	5	1	PROFILES.DBF
PARENT_UNI	C	60	0	POLEST.DBF
PHASE	N	1	0	AMMOEST.DBF
				ORDERTAS.DBF
				CL_OTHER.DBF
				POLEST.DBF
PHASE1LEN	N	5	2	ORDERLST.DBF
PHASE2LEN	N	5	2	ORDERLST.DBF
PHASE3LEN	N	5	2	ORDERLST.DBF
PHASE4LEN	N	5	2	ORDERLST.DBF
PHASE5LEN	N	5	2	ORDERLST.DBF
PHASE6LEN	N	5	2	ORDERLST.DBF
PHASE7LEN	N	5	2	ORDERLST.DBF
PHASE8LEN	N	5	2	ORDERLST.DBF
PHASE9LEN	N	5	2	ORDERLST.DBF
PHASES	N	1	0	ORDERLST.DBF
PLAN_FACT	N	5	3	CLASS_I.DBF

PLAN_FACT	N	4	2	MEDICAL.DBF
POL_QTY	N	9	2	POLEST.DBF
POL_RATE	L	1	0	LINLIST.DBF
POL_TYPE	C	3	0	POLRATE.DBF
				POLWGT.DBF
				POLEST.DBF
POSTURE	C	25	0	POSTURE.DBF
				CPROFILE.DBF
PROFILE	C	15	0	PROFILES.DBF
				CPROFILE.DBF
PROFILEDES	C	50	0	CPROFILE.DBF
PROFILENAM	C	20	0	CL_OTHER.DBF
				CPROFILE.DBF
				ORDERTAS.DBF
PROGRAM	C	8	0	HELP.DBF
QTY	N	2	0	AMMOEST.DBF
				POLEST.DBF
				TEMP.DBF
				CSRCLIST.DBF
QTY	N	4	0	LINLIST.DBF
QTY	N	2	0	TASKORG.DBF
				CL_OTHER.DBF
				SRCLIST.DBF
QUANTITY	N	6	0	TTEMP.DBF
				AMMOEST.DBF
				TTTEMP.DBF
				SRCLINQT.DBF
				CSRCLINQ.DBF
				POLEST.DBF
RATION	C	10	0	CLASS_I.DBF
SG_HRS	N	5	1	PROFILES.DBF
SR	N	8	3	POLRATE.DBF
SRC	C	9	0	CSRCLIST.DBF
				CSRCLINQ.DBF
				TTTEMP.DBF
				SRCLINQT.DBF
				POLEST.DBF
				TEMP.DBF
				CL_OTHER.DBF
				TASKORG.DBF
				TTEMP.DBF
				AMMOEST.DBF
				SRCLIST.DBF
STRENGTH	N	6	0	CSRCLIST.DBF
STRENGTH	N	3	0	ORDERTAS.DBF
STRENGTH	N	6	0	SRCLIST.DBF
				TEMP.DBF
				CL_OTHER.DBF
SUBJECT	C	60	0	HELP.DBF
SUBORDINAT	L	1	0	TEMP.DBF
				CSRCLIST.DBF
SV_HRS	N	5	1	PROFILES.DBF
TASKORG	L	1	0	TEMP.DBF
				CSRCLIST.DBF
TASKORGC	L	1	0	CSRCLIST.DBF
				TEMP.DBF
TASKORGNAM	C	60	0	TASKORG.DBF
				POLEST.DBF
				CL_OTHER.DBF

				ORDERTAS.DBF
				AMMOEST.DBF
TASKORG_CB	N	11	0	DODIC.DBF
TASKORG_QT	N	11	0	DODIC.DBF
TASK_ORG_Q	N	9	0	CLASS_I.DBF
TV_CC_HRS	N	5	1	PROFILES.DBF
TV_SR_HRS	N	5	1	PROFILES.DBF
TV_TI_HRS	N	5	1	PROFILES.DBF
UNIT	C	12	0	CLASS_I.DBF
UNITNAME	C	60	0	AMMOEST.DBF
				CL_OTHER.DBF
				TEMP.DBF
				CSRCLIST.DBF
				POLEST.DBF
				CSRCLINQ.DBF
				TASKORG.DBF
				TTEMP.DBF
UNIT_CB	N	11	0	DODIC.DBF
UNIT_DESIG	C	38	0	CSRCLIST.DBF
				TEMP.DBF
				SRCLIST.DBF
UNIT_QT	N	11	0	DODIC.DBF
UNIT_QTY	N	9	0	CLASS_I.DBF
WATER_NAME	C	19	0	WATER.DBF
				CPROFILE.DBF
WEIGHT	N	10	2	DODIC.DBF
				AMMORATE.DBF
WV_KMS	N	5	1	PROFILES.DBF

----

34 files containing procedures in the system
    INSTRUCT.PRG
    SURE.PRG
    INSTUNIT.PRG
    CREATESR.PRG
    EDITUNIT.PRG
    CUSTUNIT.PRG
    DELUNIT.PRG
    INSTTASK.PRG
    CREATETA.PRG
    CREATETC.PRG
    EDITTASK.PRG
    DELTASK.PRG
    INSTPROF.PRG
    CREATEPR.PRG
    EDITPROF.PRG
    DELPROF.PRG
    CRECUSPR.PRG
    INSTORDE.PRG
    CREATEOR.PRG
    EDITORDE.PRG
    DELORDE.PRG
    INSTREPT.PRG
    ORDEREP1.PRG
    INSTSYSR.PRG
    SYSREP.PRG
    RESTOREF.PRG
    INSTPHLP.PRG
    PRNTHELP.PRG
    PRINTPOL.PRG
    PRINTAMM.PRG
    PRINTOTH.PRG
    CALCPOL.PRG
    CALCAMMO.PRG
    CALCOTHE.PRG

INSTRUCT.PRG -- Last updated: 04/10/94 at 19:44

    Contains: SURECSR1   (Params: none)
    Contains: SURECSR2   (Params: none)
    Contains: SURECSR3   (Params: none)
    Contains: SURECSR5   (Params: none)
    Contains: SURETAS1   (Params: none)
    Contains: SURETAS2   (Params: none)
    Contains: SURETAS3   (Params: none)
    Contains: SURETAS5   (Params: none)
    Contains: SUREPRO1   (Params: none)
    Contains: SUREPRO2   (Params: none)
    Contains: SUREPRO4   (Params: none)
    Contains: SUREPRO6   (Params: none)
    Contains: SUREORD1   (Params: none)
    Contains: SUREORD2   (Params: none)
    Contains: SUREORD4   (Params: none)

```
Contains: SUREREP1                      (Params: none)
Contains: SUREREP2                      (Params: none)
Contains: SUREUTI1                      (Params: none)
Contains: SUREUTI2                      (Params: none)
Contains: SUREUTI3                      (Params: none)

Contains: SUREUTI4                      (Params: none)
Contains: SUREUTI5                      (Params: none)
Contains: SUREUTI6                      (Params: none)
Contains: SUREQUI1                      (Params: none)
Contains: SUREQUI2                      (Params: none)
Contains: SUREHEL1                      (Params: none)
Contains: SUREHEL2                      (Params: none)


--------------------------------------------------------------------

SURE.PRG -- Last updated:   04/10/94 at 19:44

    Contains: DEF_MENS                  (Params: none)
        Called by: SURE.PRG
            Calls: SRC_PRO              (procedure in SURE.PRG)
            Calls: TASK_PRO             (procedure in SURE.PRG)
            Calls: PROF_PRO             (procedure in SURE.PRG)
            Calls: ORDE_PRO             (procedure in SURE.PRG)
            Calls: REPO_PRO             (procedure in SURE.PRG)
            Calls: UTIL_PRO             (procedure in SURE.PRG)
            Calls: HELP_PRO             (procedure in SURE.PRG)
            Calls: QUIT_PRO             (procedure in SURE.PRG)
    Contains: SRC_PRO                   (Params: none)
        Called by: DEF_MENS             (procedure in SURE.PRG)
            Calls: CREATESR.PRG
            Calls: STARTHLP.PRG
            Calls: EDITUNIT.PRG
            Calls: CUSTUNIT.PRG
            Calls: DELUNIT.PRG
    Contains: TASK_PRO                  (Params: none)
        Called by: DEF_MENS             (procedure in SURE.PRG)
            Calls: CREATETA.PRG
            Calls: STARTHLP.PRG
            Calls: CREATETC.PRG
            Calls: EDITTASK.PRG
            Calls: DELTASK.PRG
    Contains: PROF_PRO                  (Params: none)
        Called by: DEF_MENS             (procedure in SURE.PRG)
            Calls: CREATEPR.PRG
            Calls: STARTHLP.PRG
            Calls: EDITPROF.PRG
            Calls: DELPROF.PRG
            Calls: CRECUSPR.PRG
    Contains: ORDE_PRO                  (Params: none)
        Called by: DEF_MENS             (procedure in SURE.PRG)
            Calls: CREATEOR.PRG
            Calls: STARTHLP.PRG
            Calls: EDITORDE.PRG
            Calls: DELORDE.PRG
    Contains: REPO_PRO                  (Params: none)
        Called by: DEF_MENS             (procedure in SURE.PRG)
            Calls: ORDEREP1.PRG
            Calls: STARTHLP.PRG
            Calls: SYSREP.PRG

                        400
```

```
       Contains: UTIL_PRO                (Params: none)
          Called by: DEF_MENS            (procedure in SURE.PRG)
                Calls: REINDX.PRG
                Calls: STARTHLP.PRG
                Calls: BACKDBF.PRG
                Calls: RESTOREF.PRG
                Calls: PRNSETUP.PRG
                Calls: COMPRESS.PRG
                Calls: UNCOMPRE.PRG
       Contains: QUIT_PRO                (Params: none)
          Called by: DEF_MENS            (procedure in SURE.PRG)
                Calls: CLOSEALL.PRG
       Contains: HELP_PRO                (Params: none)
          Called by: DEF_MENS            (procedure in SURE.PRG)
                Calls: HELPNDX.PRG
                Calls: STARTHLP.PRG
                Calls: PRNTHELP.PRG
                Calls: EDITHELP.PRG


-----------------------------------------------------------------
INSTUNIT.PRG -- Last updated:  04/10/94 at 19:45

       Contains: CSRC_BAR1               (Params: none)
       Contains: CSRC_BAR2               (Params: none)
       Contains: CSRC_BAR4               (Params: none)
       Contains: CSRC_BAR5               (Params: none)
       Contains: CSRC_BAR6               (Params: none)
       Contains: CSRC_BAR3               (Params: none)
       Contains: EDIU_BAR1               (Params: none)
       Contains: EDIU_BAR2               (Params: none)
       Contains: EDIU_BAR3               (Params: none)
       Contains: EDIU_BAR4               (Params: none)
       Contains: EDIU_BAR5               (Params: none)
       Contains: EDIU_BAR6               (Params: none)
       Contains: EDIU_BAR7               (Params: none)
       Contains: EDIU_BAR8               (Params: none)
       Contains: CUST_BAR1               (Params: none)
       Contains: CUST_BAR2               (Params: none)
       Contains: CUST_BAR3               (Params: none)
       Contains: CUST_BAR4               (Params: none)
       Contains: CUST_BAR5               (Params: none)
       Contains: CUST_BAR6               (Params: none)
       Contains: DELU_BAR1               (Params: none)
       Contains: DELU_BAR2               (Params: none)
       Contains: DELU_BAR3               (Params: none)
       Contains: DELU_BAR4               (Params: none)


-----------------------------------------------------------------
CREATESR.PRG -- Last updated:  04/16/94 at 10:55

       Contains: CSRC_PRO                (Params: none)
          Called by: CREATESR.PRG
                Calls: COPYSRC            (procedure in CREATESR.PRG)
                Calls: SEEKSRC            (procedure in CREATESR.PRG)
                Calls: CSRCENDA           (procedure in CREATESR.PRG)
                Calls: COPYCSRC           (procedure in CREATESR.PRG)
                Calls: VIEWUNIT           (procedure in CREATESR.PRG)
```

```
            Calls: CSRCEND              (procedure in CREATESR.PRG)
Contains: SEEKSRC                       (Params: none)
    Called by: CSRC_PRO                 (procedure in CREATESR.PRG)
        Calls: COPYSRC                  (procedure in CREATESR.PRG)
Contains: COPYSRC                       (Params: none)
    Called by: CSRC_PRO                 (procedure in CREATESR.PRG)
        Called by: SEEKSRC              (procedure in CREATESR.PRG)
            Calls: SETBROWS.PRG
            Calls: ENDBROWS.PRG
            Calls: DISPUNIT             (procedure in CREATESR.PRG)
Contains: COPYCSRC                      (Params: none)
    Called by: CSRC_PRO                 (procedure in CREATESR.PRG)
        Calls: SETBROWS.PRG
        Calls: ENDBROWS.PRG
        Calls: DISPUNIT                 (procedure in CREATESR.PRG)
Contains: VIEWUNIT                      (Params: none)
    Called by: CSRC_PRO                 (procedure in CREATESR.PRG)
        Calls: SETBROWS.PRG
        Calls: ENDBROWS.PRG
Contains: CSRCEND                       (Params: none)
    Called by: CSRC_PRO                 (procedure in CREATESR.PRG)
        Calls: ADDDATA                  (procedure in CREATESR.PRG)
        Calls: COPYLINS                 (procedure in CREATESR.PRG)
Contains: CSRCENDA                      (Params: none)
    Called by: CSRC_PRO                 (procedure in CREATESR.PRG)
        Calls: ADDDATA                  (procedure in CREATESR.PRG)
        Calls: COPYLINS                 (procedure in CREATESR.PRG)
        Calls: STARTCSR                 (procedure in CREATESR.PRG)
Contains: STARTCSR                      (Params: none)
    Called by: CREATESR.PRG
    Called by: CSRCENDA                 (procedure in CREATESR.PRG)
        Calls: STARTHLP.PRG
Contains: COPYLINS                      (Params: none)
    Called by: CSRCEND                  (procedure in CREATESR.PRG)
    Called by: CSRCENDA                 (procedure in CREATESR.PRG)
Contains: DISPUNIT                      (Params: none)
    Called by: COPYSRC                  (procedure in CREATESR.PRG)
    Called by: COPYCSRC                 (procedure in CREATESR.PRG)
    Called by: ADDDATA                  (procedure in CREATESR.PRG)
    Called by: SELECTUNI                (procedure in EDITUNIT.PRG)
    Called by: SELECTUN2                (procedure in EDITUNIT.PRG)
    Called by: CHANGELI                 (procedure in EDITUNIT.PRG)
    Called by: ADDLINCA                 (procedure in EDITUNIT.PRG)
    Called by: DELLIN                   (procedure in EDITUNIT.PRG)
    Called by: CHANGEEH                 (procedure in EDITUNIT.PRG)
    Called by: EDITSTR                  (procedure in EDITUNIT.PRG)
    Called by: UNITDATA                 (procedure in CUSTUNIT.PRG)
Contains: ADDDATA                       (Params: none)
    Called by: CSRCEND                  (procedure in CREATESR.PRG)
    Called by: CSRCENDA                 (procedure in CREATESR.PRG)
        Calls: ECHELON.PRG
        Calls: SETBROWS.PRG
        Calls: ENDBROWS.PRG
        Calls: DISPUNIT                 (procedure in CREATESR.PRG)
```

EDITUNIT.PRG -- Last updated: 04/10/94 at 19:44

```
Contains: EDIU_PRO                    (Params: none)
    Called by: EDITUNIT.PRG
        Calls: SELECTUNI              (procedure in EDITUNIT.PRG)
        Calls: SELECTUN2              (procedure in EDITUNIT.PRG)
        Calls: CHANGELI               (procedure in EDITUNIT.PRG)
        Calls: ADDLINCA               (procedure in EDITUNIT.PRG)
        Calls: DELLIN                 (procedure in EDITUNIT.PRG)
        Calls: EDITSTR                (procedure in EDITUNIT.PRG)
        Calls: CHANGEEH               (procedure in EDITUNIT.PRG)
        Calls: ENDUNIT                (procedure in EDITUNIT.PRG)
Contains: STARTUNI                    (Params: none)
    Called by: EDITUNIT.PRG
    Called by: CUSTUNIT.PRG
        Calls: STARTHLP.PRG
        Calls: HELP.PRG
        Calls: ENDUNIT                (procedure in EDITUNIT.PRG)
Contains: SELECTUNI                   (Params: none)
    Called by: EDIU_PRO               (procedure in EDITUNIT.PRG)
    Called by: DELU_PRO               (procedure in DELUNIT.PRG)
        Calls: SETBROWS.PRG
        Calls: ENDBROWS.PRG
        Calls: DISPUNIT               (procedure in CREATESR.PRG)
Contains: SELECTUN2                   (Params: none)
    Called by: EDIU_PRO               (procedure in EDITUNIT.PRG)
    Called by: DELU_PRO               (procedure in DELUNIT.PRG)
        Calls: SETBROWS.PRG
        Calls: ENDBROWS.PRG
        Calls: DISPUNIT               (procedure in CREATESR.PRG)
Contains: CHANGELI                    (Params: none)
    Called by: EDIU_PRO               (procedure in EDITUNIT.PRG)
    Called by: CUST_PRO               (procedure in CUSTUNIT.PRG)
        Calls: SETBROWS.PRG
        Calls: ENDBROWS.PRG
        Calls: DISPUNIT               (procedure in CREATESR.PRG)
Contains: ADDLINCA                    (Params: none)
    Called by: EDIU_PRO               (procedure in EDITUNIT.PRG)
    Called by: CUST_PRO               (procedure in CUSTUNIT.PRG)
        Calls: SETBROWS.PRG
        Calls: ENDBROWS.PRG
        Calls: DISPUNIT               (procedure in CREATESR.PRG)
Contains: DELLIN                      (Params: none)
    Called by: EDIU_PRO               (procedure in EDITUNIT.PRG)
    Called by: CUST_PRO               (procedure in CUSTUNIT.PRG)
        Calls: SETBROWS.PRG
        Calls: ENDBROWS.PRG
        Calls: DISPUNIT               (procedure in CREATESR.PRG)
Contains: CHANGEEH                    (Params: none)
    Called by: EDIU_PRO               (procedure in EDITUNIT.PRG)
    Called by: CUST_PRO               (procedure in CUSTUNIT.PRG)
        Calls: SETBROWS.PRG
        Calls: ENDBROWS.PRG
        Calls: DISPUNIT               (procedure in CREATESR.PRG)
Contains: ENDUNIT                     (Params: none)
    Called by: STARTUNI               (procedure in EDITUNIT.PRG)
    Called by: EDIU_PRO               (procedure in EDITUNIT.PRG)
    Called by: CUST_PRO               (procedure in CUSTUNIT.PRG)
Contains: EDITSTR                     (Params: none)
    Called by: EDIU_PRO               (procedure in EDITUNIT.PRG)
        Calls: DISPUNIT               (procedure in CREATESR.PRG)
```

```
    Contains: DISPUNIT                  (Params: none)
       Called by: COPYSRC               (procedure in CREATESR.PRG)
       Called by: COPYCSRC              (procedure in CREATESR.PRG)
       Called by: ADDDATA               (procedure in CREATESR.PRG)
       Called by: SELECTUNI             (procedure in EDITUNIT.PRG)
       Called by: SELECTUN2             (procedure in EDITUNIT.PRG)
       Called by: CHANGELI              (procedure in EDITUNIT.PRG)
       Called by: ADDLINCA              (procedure in EDITUNIT.PRG)
       Called by: DELLIN                (procedure in EDITUNIT.PRG)
       Called by: CHANGEEH              (procedure in EDITUNIT.PRG)
       Called by: EDITSTR               (procedure in EDITUNIT.PRG)
       Called by: UNITDATA              (procedure in CUSTUNIT.PRG)


-------------------------------------------------------------------
CUSTUNIT.PRG -- Last updated:  04/10/94 at 19:44

    Contains: CUST_PRO                  (Params: none)
       Called by: CUSTUNIT.PRG
          Calls: UNITDATA               (procedure in CUSTUNIT.PRG)
          Calls: CHANGELI               (procedure in EDITUNIT.PRG)
          Calls: ADDLINCA               (procedure in EDITUNIT.PRG)
          Calls: DELLIN                 (procedure in EDITUNIT.PRG)
          Calls: CHANGEEH               (procedure in EDITUNIT.PRG)
          Calls: ENDUNIT                (procedure in EDITUNIT.PRG)
    Contains: STARTUNI                  (Params: none)
       Called by: EDITUNIT.PRG
       Called by: CUSTUNIT.PRG
          Calls: STARTHLP.PRG
          Calls: HELP.PRG
          Calls: ENDUNIT                (procedure in EDITUNIT.PRG)
    Contains: UNITDATA                  (Params: none)
       Called by: CUST_PRO              (procedure in CUSTUNIT.PRG)
          Calls: ECHELON.PRG
          Calls: DISPUNIT               (procedure in CREATESR.PRG)
    Contains: CHANGELI                  (Params: none)
       Called by: EDIU_PRO              (procedure in EDITUNIT.PRG)
       Called by: CUST_PRO              (procedure in CUSTUNIT.PRG)
          Calls: SETBROWS.PRG
          Calls: ENDBROWS.PRG
          Calls: DISPUNIT               (procedure in CREATESR.PRG)
    Contains: ADDLINCA                  (Params: none)
       Called by: EDIU_PRO              (procedure in EDITUNIT.PRG)
       Called by: CUST_PRO              (procedure in CUSTUNIT.PRG)
          Calls: SETBROWS.PRG
          Calls: ENDBROWS.PRG
          Calls: DISPUNIT               (procedure in CREATESR.PRG)
    Contains: DELLIN                    (Params: none)
       Called by: EDIU_PRO              (procedure in EDITUNIT.PRG)
       Called by: CUST_PRO              (procedure in CUSTUNIT.PRG)
          Calls: SETBROWS.PRG
          Calls: ENDBROWS.PRG
          Calls: DISPUNIT               (procedure in CREATESR.PRG)
    Contains: CHANGEEH                  (Params: none)
       Called by: EDIU_PRO              (procedure in EDITUNIT.PRG)
       Called by: CUST_PRO              (procedure in CUSTUNIT.PRG)
          Calls: SETBROWS.PRG
          Calls: ENDBROWS.PRG
          Calls: DISPUNIT               (procedure in CREATESR.PRG)
```

```
    Contains: ENDUNIT                      (Params: none)
        Called by: STARTUNI                (procedure in EDITUNIT.PRG)
        Called by: EDIU_PRO                (procedure in EDITUNIT.PRG)
        Called by: CUST_PRO                (procedure in CUSTUNIT.PRG)
    Contains: DISPUNIT                      (Params: none)
        Called by: COPYSRC                 (procedure in CREATESR.PRG)
        Called by: COPYCSRC                (procedure in CREATESR.PRG)
        Called by: ADDDATA                 (procedure in CREATESR.PRG)
        Called by: SELECTUNI               (procedure in EDITUNIT.PRG)
        Called by: SELECTUN2               (procedure in EDITUNIT.PRG)
        Called by: CHANGELI                (procedure in EDITUNIT.PRG)
        Called by: ADDLINCA                (procedure in EDITUNIT.PRG)
        Called by: DELLIN                  (procedure in EDITUNIT.PRG)
        Called by: CHANGEEH                (procedure in EDITUNIT.PRG)
        Called by: EDITSTR                 (procedure in EDITUNIT.PRG)
        Called by: UNITDATA                (procedure in CUSTUNIT.PRG)


-----------------------------------------------------------------

DELUNIT.PRG -- Last updated:   04/10/94 at 19:44

    Contains: DELU_PRO                      (Params: none)
        Called by: DELUNIT.PRG
            Calls: SELECTUNI               (procedure in EDITUNIT.PRG)
            Calls: SELECTUN2               (procedure in EDITUNIT.PRG)
            Calls: DELUAS                  (procedure in DELUNIT.PRG)
            Calls: ENDDELU                 (procedure in DELUNIT.PRG)
    Contains: ENDDELU                       (Params: none)
        Called by: DELU_PRO                (procedure in DELUNIT.PRG)
    Contains: DELUAS                        (Params: none)
        Called by: DELU_PRO                (procedure in DELUNIT.PRG)
    Contains: STARTED                       (Params: none)
        Called by: DELUNIT.PRG
        Called by: EDITTASK.PRG
        Called by: DELTASK.PRG
        Called by: DELPROF.PRG
            Calls: STARTHLP.PRG
    Contains: SELECTUNI                      (Params: none)
        Called by: EDIU_PRO                (procedure in EDITUNIT.PRG)
        Called by: DELU_PRO                (procedure in DELUNIT.PRG)
            Calls: SETBROWS.PRG
            Calls: ENDBROWS.PRG
            Calls: DISPUNIT                (procedure in CREATESR.PRG)
    Contains: SELECTUN2                      (Params: none)
        Called by: EDIU_PRO                (procedure in EDITUNIT.PRG)
        Called by: DELU_PRO                (procedure in DELUNIT.PRG)
            Calls: SETBROWS.PRG
            Calls: ENDBROWS.PRG
            Calls: DISPUNIT                (procedure in CREATESR.PRG)
    Contains: DISPUNIT                       (Params: none)
        Called by: COPYSRC                 (procedure in CREATESR.PRG)
        Called by: COPYCSRC                (procedure in CREATESR.PRG)
        Called by: ADDDATA                 (procedure in CREATESR.PRG)
        Called by: SELECTUNI               (procedure in EDITUNIT.PRG)
        Called by: SELECTUN2               (procedure in EDITUNIT.PRG)
        Called by: CHANGELI                (procedure in EDITUNIT.PRG)
        Called by: ADDLINCA                (procedure in EDITUNIT.PRG)
        Called by: DELLIN                  (procedure in EDITUNIT.PRG)
        Called by: CHANGEEH                (procedure in EDITUNIT.PRG)
```

```
        Called by: EDITSTR           (procedure in EDITUNIT.PRG)
        Called by: UNITDATA          (procedure in CUSTUNIT.PRG)


-----------------------------------------------------------------
INSTTASK.PRG -- Last updated:  04/16/94 at 10:54

    Contains: CRET_BAR1             (Params: none)
    Contains: CRET_BAR2             (Params: none)
    Contains: CRET_BAR3             (Params: none)
    Contains: CRET_BAR4             (Params: none)
    Contains: CRET_BAR5             (Params: none)
    Contains: CRET_BAR6             (Params: none)
    Contains: COPT_BAR1             (Params: none)
    Contains: COPT_BAR2             (Params: none)
    Contains: COPT_BAR4             (Params: none)
    Contains: COPT_BAR3             (Params: none)
    Contains: EDIT_BAR1             (Params: none)
    Contains: EDIT_BAR2             (Params: none)
    Contains: EDIT_BAR3             (Params: none)
    Contains: EDIT_BAR4             (Params: none)
    Contains: EDIT_BAR5             (Params: none)
    Contains: EDIT_BAR6             (Params: none)
    Contains: EDIT_BAR7             (Params: none)
    Contains: EDIT_BAR8             (Params: none)
    Contains: DELT_BAR1             (Params: none)
    Contains: DELT_BAR2             (Params: none)
    Contains: DELT_BAR3             (Params: none)


-----------------------------------------------------------------
CREATETA.PRG -- Last updated:  04/15/94 at 15:21

    Contains: CRET_PRO              (Params: none)
        Called by: CREATETA.PRG
            Calls: NAMETA            (procedure in CREATETA.PRG)
            Calls: ASSIGNUN          (procedure in CREATETA.PRG)
            Calls: ASSIGNU2          (procedure in CREATETA.PRG)
            Calls: UNITQTY           (procedure in CREATETA.PRG)
            Calls: ENDTASKA          (procedure in CREATETA.PRG)
            Calls: ENDTASK           (procedure in CREATETA.PRG)
    Contains: NAMETA                (Params: none)
        Called by: CRET_PRO          (procedure in CREATETA.PRG)
            Calls: DISPTASK          (procedure in CREATETA.PRG)
    Contains: ASSIGNUN              (Params: none)
        Called by: CRET_PRO          (procedure in CREATETA.PRG)
        Called by: EDIT_PRO          (procedure in EDITTASK.PRG)
            Calls: DISPTASK          (procedure in CREATETA.PRG)
    Contains: ASSIGNU2              (Params: none)
        Called by: CRET_PRO          (procedure in CREATETA.PRG)
        Called by: EDIT_PRO          (procedure in EDITTASK.PRG)
            Calls: DISPTASK          (procedure in CREATETA.PRG)
    Contains: UNITQTY               (Params: none)
        Called by: CRET_PRO          (procedure in CREATETA.PRG)
    Contains: ENDTASK               (Params: none)
        Called by: CRET_PRO          (procedure in CREATETA.PRG)
    Contains: ENDTASKA              (Params: none)
        Called by: CRET_PRO          (procedure in CREATETA.PRG)
    Contains: STARTASK              (Params: none)
```

```
                Called by: CREATETA.PRG
                Called by: CREATETC.PRG
                    Calls: STARTHLP.PRG
            Contains: DISPTASK                (Params: none)
                Called by: NAMETA             (procedure in CREATETA.PRG)
                Called by: ASSIGNUN           (procedure in CREATETA.PRG)
                Called by: ASSIGNU2           (procedure in CREATETA.PRG)
                Called by: FINDTA             (procedure in CREATETC.PRG)
                Called by: ASSIGNNA           (procedure in CREATETC.PRG)
                Called by: VIEWTASK           (procedure in EDITTASK.PRG)
                Called by: VIEWTAS2           (procedure in EDITTASK.PRG)
                Called by: UNASSIGU           (procedure in EDITTASK.PRG)
                Called by: UNASSIG2           (procedure in EDITTASK.PRG)
            Contains: DISPUNI                 (Params: none)


------------------------------------------------------------------------

CREATETC.PRG -- Last updated:  04/10/94 at 19:44


            Contains: COPT_PRO                (Params: none)
                Called by: CREATETC.PRG
                    Calls: FINDTA             (procedure, in CREATETC.PRG)
                    Calls: ASSIGNNA           (procedure in CREATETC.PRG)
                    Calls: ENDCOPT            (procedure in CREATETC.PRG)
                    Calls: ENDCOPTA           (procedure in CREATETC.PRG)
            Contains: STARTASK                (Params: none)
                Called by: CREATETA.PRG
                Called by: CREATETC.PRG
                    Calls: STARTHLP.PRG
            Contains: FINDTA                  (Params: none)
                Called by: EDITTASK.PRG
                Called by: COPT_PRO           (procedure in CREATETC.PRG)
                Called by: EDIT_PRO           (procedure in EDITTASK.PRG)
                Called by: DELT_PRO           (procedure in DELTASK.PRG)
                    Calls: SETBROWS.PRG
                    Calls: ENDBROWS.PRG
                    Calls: DISPTASK           (procedure in CREATETA.PRG)
            Contains: ASSIGNNA                (Params: none)
                Called by: COPT_PRO           (procedure in CREATETC.PRG)
                    Calls: DISPTASK           (procedure in CREATETA.PRG)
            Contains: ENDCOPT                 (Params: none)
                Called by: COPT_PRO           (procedure in CREATETC.PRG)
            Contains: ENDCOPTA                (Params: none)
                Called by: COPT_PRO           (procedure in CREATETC.PRG)
            Contains: DISPTASK                (Params: none)
                Called by: NAMETA             (procedure in CREATETA.PRG)
                Called by: ASSIGNUN           (procedure in CREATETA.PRG)
                Called by: ASSIGNU2           (procedure in CREATETA.PRG)
                Called by: FINDTA             (procedure in CREATETC.PRG)
                Called by: ASSIGNNA           (procedure in CREATETC.PRG)
                Called by: VIEWTASK           (procedure in EDITTASK.PRG)
                Called by: VIEWTAS2           (procedure in EDITTASK.PRG)
                Called by: UNASSIGU           (procedure in EDITTASK.PRG)
                Called by: UNASSIG2           (procedure in EDITTASK.PRG)


------------------------------------------------------------------------

EDITTASK.PRG -- Last updated:  04/12/94 at 21:43

                               407
```

```
Contains: EDIT_PRO                      (Params: none)
    Called by: EDITTASK.PRG
        Calls: FINDTA                   (procedure in CREATETC.PRG)
        Calls: ASSIGNUN                 (procedure in CREATETA.PRG)
        Calls: ASSIGNU2                 (procedure in CREATETA.PRG)
        Calls: VIEWTASK                 (procedure in EDITTASK.PRG)
        Calls: VIEWTAS2                 (procedure in EDITTASK.PRG)
        Calls: UNASSIGU                 (procedure in EDITTASK.PRG)
        Calls: UNASSIG2                 (procedure in EDITTASK.PRG)
        Calls: ENDEDIT                  (procedure in EDITTASK.PRG)
Contains: FINDTA                        (Params: none)
    Called by: EDITTASK.PRG
    Called by: COPT_PRO                 (procedure in CREATETC.PRG)
    Called by: EDIT_PRO                 (procedure in EDITTASK.PRG)
    Called by: DELT_PRO                 (procedure in DELTASK.PRG)
        Calls: SETBROWS.PRG
        Calls: ENDBROWS.PRG
        Calls: DISPTASK                 (procedure in CREATETA.PRG)
Contains: ASSIGNUN                      (Params: none)
    Called by: CRET_PRO                 (procedure in CREATETA.PRG)
    Called by: EDIT_PRO                 (procedure in EDITTASK.PRG)
        Calls: DISPTASK                 (procedure in CREATETA.PRG)
Contains: ASSIGNU2                      (Params: none)
    Called by: CRET_PRO                 (procedure in CREATETA.PRG)
    Called by: EDIT_PRO                 (procedure in EDITTASK.PRG)
        Calls: DISPTASK                 (procedure in CREATETA.PRG)
Contains: VIEWTASK                      (Params: none)
    Called by: EDIT_PRO                 (procedure in EDITTASK.PRG)
        Calls: SETBROW2.PRG
        Calls: ENDBROWS.PRG
        Calls: DISPTASK                 (procedure in CREATETA.PRG)
Contains: VIEWTAS2                      (Params: none)
    Called by: EDIT_PRO                 (procedure in EDITTASK.PRG)
        Calls: SETBROW2.PRG
        Calls: ENDBROWS.PRG
        Calls: DISPTASK                 (procedure in CREATETA.PRG)
Contains: UNASSIGU                      (Params: none)
    Called by: EDIT_PRO                 (procedure in EDITTASK.PRG)
        Calls: DISPTASK                 (procedure in CREATETA.PRG)
Contains: UNASSIG2                      (Params: none)
    Called by: EDIT_PRO                 (procedure in EDITTASK.PRG)
        Calls: DISPTASK                 (procedure in CREATETA.PRG)
Contains: ENDEDIT                       (Params: none)
    Called by: EDIT_PRO                 (procedure in EDITTASK.PRG)
Contains: STARTED                       (Params: none)
    Called by: DELUNIT.PRG
    Called by: EDITTASK.PRG
    Called by: DELTASK.PRG
    Called by: DELPROF.PRG
        Calls: STARTHLP.PRG
Contains: DISPTASK                      (Params: none)
    Called by: NAMETA                   (procedure in CREATETA.PRG)
    Called by: ASSIGNUN                 (procedure in CREATETA.PRG)
    Called by: ASSIGNU2                 (procedure in CREATETA.PRG)
    Called by: FINDTA                   (procedure in CREATETC.PRG)
    Called by: ASSIGNNA                 (procedure in CREATETC.PRG)
    Called by: VIEWTASK                 (procedure in EDITTASK.PRG)
    Called by: VIEWTAS2                 (procedure in EDITTASK.PRG)
    Called by: UNASSIGU                 (procedure in EDITTASK.PRG)
```

408

```
        Called by: UNASSIG2          '     (procedure in EDITTASK.PRG)
        Contains: DISPUNI                  (Params: none)


-------------------------------------------------------------------
DELTASK.PRG -- Last updated:  04/10/94 at 19:44

        Contains: DELT_PRO                 (Params: none)
           Called by: DELTASK.PRG
              Calls: FINDTA               (procedure in CREATETC.PRG)
              Calls: DELTAS               (procedure in DELTASK.PRG)
              Calls: ENDDELT              (procedure in DELTASK.PRG)
        Contains: FINDTA                   (Params: none)
           Called by: EDITTASK.PRG
           Called by: COPT_PRO            (procedure in CREATETC.PRG)
           Called by: EDIT_PRO            (procedure in EDITTASK.PRG)
           Called by: DELT_PRO            (procedure in DELTASK.PRG)
              Calls: SETBROWS.PRG
              Calls: ENDBROWS.PRG
              Calls: DISPTASK             (procedure in CREATETA.PRG)
        Contains: ENDDELT                  (Params: none)
           Called by: DELT_PRO            (procedure in DELTASK.PRG)
        Contains: DELTAS                   (Params: none)
           Called by: DELT_PRO            (procedure in DELTASK.PRG)
        Contains: STARTED                  (Params: none)
           Called by: DELUNIT.PRG
           Called by: EDITTASK.PRG
           Called by: DELTASK.PRG
           Called by: DELPROF.PRG
              Calls: STARTHLP.PRG
        Contains: DISPTASK                 (Params: none)
           Called by: NAMETA              (procedure in CREATETA.PRG)
           Called by: ASSIGNUN            (procedure in CREATETA.PRG)
           Called by: ASSIGNU2            (procedure in CREATETA.PRG)
           Called by: FINDTA              (procedure in CREATETC.PRG)
           Called by: ASSIGNNA            (procedure in CREATETC.PRG)
           Called by: VIEWTASK            (procedure in EDITTASK.PRG)
           Called by: VIEWTAS2            (procedure in EDITTASK.PRG)
           Called by: UNASSIGU            (procedure in EDITTASK.PRG)
           Called by: UNASSIG2            (procedure in EDITTASK.PRG)


-------------------------------------------------------------------
INSTPROF.PRG -- Last updated:  04/10/94 at 19:45

        Contains: CREP_BAR1                (Params: none)
        Contains: CREP_BAR2                (Params: none)
        Contains: CREP_BAR3                (Params: none)
        Contains: EDIP_BAR1                (Params: none)
        Contains: EDIP_BAR2                (Params: none)
        Contains: EDIP_BAR3                (Params: none)
        Contains: EDIP_BAR4                (Params: none)
        Contains: EDIP_BAR5                (Params: none)
        Contains: EDIP_BAR6                (Params: none)
        Contains: EDIP_BAR7                (Params: none)
        Contains: EDIP_BAR8                (Params: none)
        Contains: EDIP_BAR9                (Params: none)
        Contains: DELP_BAR1                (Params: none)
        Contains: DELP_BAR2                (Params: none)
```

```
   Contains: DELP_BAR3           (Params: none)
   Contains: CRCP_BAR1           (Params: none)
   Contains: CRCP_BAR2           (Params: none)
   Contains: CRCP_BAR3           (Params: none)
   Contains: CRCP_BAR4           (Params: none)
   Contains: CRCP_BAR6           (Params: none)


-----------------------------------------------------------------
CREATEPR.PRG -- Last updated:  04/10/94 at 19:44

   Contains: CREP_PRO            (Params: none)
      Called by: CREATEPR.PRG
         Calls: NEWPROFI         (procedure in CREATEPR.PRG)
         Calls: VIEWPROF         (procedure in CREATEPR.PRG)
         Calls: ENDCREP          (procedure in CREATEPR.PRG)
   Contains: NEWPROFI            (Params: none)
      Called by: CREP_PRO        (procedure in CREATEPR.PRG)
         Calls: SETBROWS.PRG
         Calls: ENDBROWS.PRG
         Calls: DISPPROF         (procedure in CREATEPR.PRG)
   Contains: VIEWPROF            (Params: none)
      Called by: CREP_PRO        (procedure in CREATEPR.PRG)
         Calls: SETBROWS.PRG
         Calls: ENDBROWS.PRG
         Calls: DISPPROF         (procedure in CREATEPR.PRG)
   Contains: ENDCREP             (Params: none)
      Called by: CREP_PRO        (procedure in CREATEPR.PRG)
      Called by: EDIP_PRO        (procedure in EDITPROF.PRG)
   Contains: STARTPRO            (Params: none)
      Called by: CREATEPR.PRG
      Called by: CRECUSPR.PRG
         Calls: STARTHLP.PRG
   Contains: DISPPROF            (Params: none)
      Called by: NEWPROFI        (procedure in CREATEPR.PRG)
      Called by: VIEWPROF        (procedure in CREATEPR.PRG)
      Called by: SELEPROF        (procedure in EDITPROF.PRG)
      Called by: CHANGEAP        (procedure in EDITPROF.PRG)
      Called by: CHANGEAC        (procedure in EDITPROF.PRG)
      Called by: CHANGEI         (procedure in EDITPROF.PRG)
      Called by: CHANGEIF        (procedure in EDITPROF.PRG)
      Called by: CHANGECI        (procedure in EDITPROF.PRG)
      Called by: CHANGEH2        (procedure in EDITPROF.PRG)
      Called by: CHANGEME        (procedure in EDITPROF.PRG)
      Called by: SELECTPRO       (procedure in DELPROF.PRG)
      Called by: FINDPRO         (procedure in CREATEOR.PRG)


-----------------------------------------------------------------
EDITPROF.PRG -- Last updated:  04/10/94 at 19:44

   Contains: EDIP_PRO            (Params: none)
      Called by: EDITPROF.PRG
         Calls: SELEPROF         (procedure in EDITPROF.PRG)
         Calls: CHANGEAP         (procedure in EDITPROF.PRG)
         Calls: CHANGEAC         (procedure in EDITPROF.PRG)
         Calls: CHANGEI          (procedure in EDITPROF.PRG)
         Calls: CHANGEIF         (procedure in EDITPROF.PRG)
         Calls: CHANGECI         (procedure in EDITPROF.PRG)
```

410

```
                Calls: CHANGEH2              (procedure in EDITPROF.PRG)
                Calls: CHANGEME              (procedure in EDITPROF.PRG)
                Calls: ENDCREP               (procedure in CREATEPR.PRG)
Contains: SELEPROF                           (Params: none)
    Called by: EDIP_PRO                      (procedure in EDITPROF.PRG)
                Calls: SETBROWS.PRG
                Calls: ENDBROWS.PRG
                Calls: DISPPROF              (procedure in CREATEPR.PRG)
Contains: CHANGEAP                           (Params: none)
    Called by: EDIP_PRO                      (procedure in EDITPROF.PRG)
                Calls: SETBROWS.PRG
                Calls: ENDBROWS.PRG
                Calls: DISPPROF              (procedure in CREATEPR.PRG)
Contains: CHANGEAC                           (Params: none)
    Called by: EDIP_PRO                      (procedure in EDITPROF.PRG)
                Calls: SETBROWS.PRG
                Calls: ENDBROWS.PRG
                Calls: DISPPROF              (procedure in CREATEPR.PRG)
Contains: CHANGEI                            (Params: none)
    Called by: EDIP_PRO                      (procedure in EDITPROF.PRG)
                Calls: SETBROWS.PRG
                Calls: ENDBROWS.PRG
                Calls: DISPPROF              (procedure in CREATEPR.PRG)
Contains: CHANGEIF                           (Params: none)
    Called by: EDIP_PRO                      (procedure in EDITPROF.PRG)
                Calls: DISPPROF              (procedure in CREATEPR.PRG)
Contains: CHANGECI                           (Params: none)
    Called by: EDIP_PRO                      (procedure in EDITPROF.PRG)
                Calls: SETBROWS.PRG
                Calls: ENDBROWS.PRG
                Calls: DISPPROF              (procedure in CREATEPR.PRG)
Contains: CHANGEH2                           (Params: none)
    Called by: EDIP_PRO                      (procedure in EDITPROF.PRG)
                Calls: SETBROWS.PRG
                Calls: ENDBROWS.PRG
                Calls: DISPPROF              (procedure in CREATEPR.PRG)
Contains: CHANGEME                           (Params: none)
    Called by: EDIP_PRO                      (procedure in EDITPROF.PRG)
                Calls: SETBROWS.PRG
                Calls: ENDBROWS.PRG
                Calls: DISPPROF              (procedure in CREATEPR.PRG)
Contains: ENDCREP                            (Params: none)
    Called by: CREP_PRO                      (procedure in CREATEPR.PRG)
    Called by: EDIP_PRO                      (procedure in EDITPROF.PRG)
Contains: STARTEPR                           (Params: none)
    Called by: EDITPROF.PRG
                Calls: STARTHLP.PRG
Contains: DISPPROF                           (Params: none)
    Called by: NEWPROFI                      (procedure in CREATEPR.PRG)
    Called by: VIEWPROF                      (procedure in CREATEPR.PRG)
    Called by: SELEPROF                      (procedure in EDITPROF.PRG)
    Called by: CHANGEAP                      (procedure in EDITPROF.PRG)
    Called by: CHANGEAC                      (procedure in EDITPROF.PRG)
    Called by: CHANGEI                       (procedure in EDITPROF.PRG)
    Called by: CHANGEIF                      (procedure in EDITPROF.PRG)
    Called by: CHANGECI                      (procedure in EDITPROF.PRG)
    Called by: CHANGEH2                      (procedure in EDITPROF.PRG)
    Called by: CHANGEME                      (procedure in EDITPROF.PRG)
    Called by: SELECTPRO                     (procedure in DELPROF.PRG)
```

411

```
------------------------------------------------------------------
```

DELPROF.PRG -- Last updated: 04/15/94 at 19:50

    Contains: DELP_PRO   (Params: none)
       Called by: DELPROF.PRG
          Calls: SELECTPRO   (procedure in DELPROF.PRG)
          Calls: DELPRO   (procedure in DELPROF.PRG)
          Calls: ENDDELP   (procedure in DELPROF.PRG)
    Contains: ENDDELP   (Params: none)
       Called by: DELP_PRO   (procedure in DELPROF.PRG)
    Contains: DELPRO   (Params: none)
       Called by: DELP_PRO   (procedure in DELPROF.PRG)
    Contains: STARTED   (Params: none)
       Called by: DELUNIT.PRG
       Called by: EDITTASK.PRG
       Called by: DELTASK.PRG
       Called by: DELPROF.PRG
          Calls: STARTHLP.PRG
    Contains: SELECTPRO   (Params: none)
       Called by: DELP_PRO   (procedure in DELPROF.PRG)
          Calls: SETBROWS.PRG
          Calls: ENDBROWS.PRG
          Calls: DISPPROF   (procedure in CREATEPR.PRG)
    Contains: DISPPROF   (Params: none)
       Called by: NEWPROFI   (procedure in CREATEPR.PRG)
       Called by: VIEWPROF   (procedure in CREATEPR.PRG)
       Called by: SELEPROF   (procedure in EDITPROF.PRG)
       Called by: CHANGEAP   (procedure in EDITPROF.PRG)
       Called by: CHANGEAC   (procedure in EDITPROF.PRG)
       Called by: CHANGEI   (procedure in EDITPROF.PRG)
       Called by: CHANGEIF   (procedure in EDITPROF.PRG)
       Called by: CHANGECI   (procedure in EDITPROF.PRG)
       Called by: CHANGEH2   (procedure in EDITPROF.PRG)
       Called by: CHANGEME   (procedure in EDITPROF.PRG)
       Called by: SELECTPRO   (procedure in DELPROF.PRG)
       Called by: FINDPRO   (procedure in CREATEOR.PRG)


```
------------------------------------------------------------------
```

CRECUSPR.PRG -- Last updated: 04/10/94 at 19:45

    Contains: CRCP_PRO   (Params: none)
       Called by: CRECUSPR.PRG
          Calls: NEWCUSPR   (procedure in CRECUSPR.PRG)
          Calls: VIEWCUSP   (procedure in CRECUSPR.PRG)
          Calls: EDITCUSP   (procedure in CRECUSPR.PRG)
          Calls: ENDCUSPR   (procedure in CRECUSPR.PRG)
          Calls: DELCUSPR   (procedure in CRECUSPR.PRG)
    Contains: NEWCUSPR   (Params: none)
       Called by: CRCP_PRO   (procedure in CRECUSPR.PRG)
          Calls: SETBROWS.PRG
          Calls: ENDBROWS.PRG
    Contains: VIEWCUSP   (Params: none)
       Called by: CRCP_PRO   (procedure in CRECUSPR.PRG)
          Calls: SETBROWS.PRG
          Calls: ENDBROWS.PRG

                              412

```
        Calls: DISPPRO            (procedure in CRECUSPR.PRG)
    Contains: EDITCUSP            (Params: none)
        Called by: CRCP_PRO       (procedure in CRECUSPR.PRG)
            Calls: SETBROWS.PRG
            Calls: ENDBROWS.PRG
    Contains: ENDCUSPR            (Params: none)
        Called by: CRCP_PRO       (procedure in CRECUSPR.PRG)
    Contains: DELCUSPR            (Params: none)
        Called by: CRCP_PRO       (procedure in CRECUSPR.PRG)
            Calls: SETBROWS.PRG
            Calls: ENDBROWS.PRG
            Calls: DISPPRO         (procedure in CRECUSPR.PRG)
    Contains: STARTPRO            (Params: none)
        Called by: CREATEPR.PRG
        Called by: CRECUSPR.PRG
            Calls: STARTHLP.PRG
    Contains: DISPPRO             (Params: none)
        Called by: VIEWCUSP        (procedure in CRECUSPR.PRG)
        Called by: DELCUSPR        (procedure in CRECUSPR.PRG)


---------------------------------------------------------------

INSTORDE.PRG -- Last updated:  04/10/94 at 19:45

    Contains: ORDS_BAR1           (Params: none)
    Contains: ORDS_BAR2           (Params: none)
    Contains: ORDS_BAR3           (Params: none)
    Contains: ORDS_BAR4           (Params: none)
    Contains: ORDS_BAR5           (Params: none)
    Contains: TOMP_BAR1           (Params: none)
    Contains: TOMP_BAR2           (Params: none)
    Contains: TOMP_BAR3           (Params: none)
    Contains: TOMP_BAR4           (Params: none)
    Contains: TOMP_BAR5           (Params: none)
    Contains: TOMP_BAR6           (Params: none)
    Contains: ORDT_BAR1           (Params: none)
    Contains: ORDT_BAR2           (Params: none)
    Contains: ORDT_BAR3           (Params: none)
    Contains: ORDT_BAR4           (Params: none)
    Contains: ORDT_BAR5           (Params: none)
    Contains: ORDT_BAR6           (Params: none)
    Contains: ORDT_BAR7           (Params: none)
    Contains: ORDT_BAR8           (Params: none)
    Contains: DORD_BAR1           (Params: none)
    Contains: DORD_BAR2           (Params: none)
    Contains: DORD_BAR3           (Params: none)
    Contains: STRE_BAR1           (Params: none)
    Contains: STRE_BAR2           (Params: none)
    Contains: STRE_BAR3           (Params: none)
    Contains: STRE_BAR4           (Params: none)
    Contains: STRE_BAR5           (Params: none)
    Contains: STRE_BAR6           (Params: none)


---------------------------------------------------------------

CREATEOR.PRG -- Last updated:  04/10/94 at 19:45

    Contains: ORDS_PRO            (Params: none)
        Called by: CREATEOR.PRG
```

```
            Calls: STARTOR       (procedure in CREATEOR.PRG)
            Calls: CREATEO       (procedure in CREATEOR.PRG)
            Calls: VIEWORDE      (procedure in CREATEOR.PRG)
            Calls: ASSIGNTA      (procedure in CREATEOR.PRG)
            Calls: ENDORDR       (procedure in CREATEOR.PRG)
Contains: CREATEO               (Params: none)
    Called by: ORDS_PRO         (procedure in CREATEOR.PRG)
        Calls: DISPORD          (procedure in CREATEOR.PRG)
Contains: VIEWORDE              (Params: none)
    Called by: ORDS_PRO         (procedure in CREATEOR.PRG)
    Called by: ORDT_PRO         (procedure in EDITORDE.PRG)
    Called by: GETORDER         (procedure in EDITORDE.PRG)
        Calls: SETBROWS.PRG
        Calls: ENDBROWS.PRG
        Calls: DISPORD2          (procedure in CREATEOR.PRG)
        Calls: DISPORD3          (procedure in ORDEREP1.PRG)
Contains: ASSIGNTA              (Params: none)
    Called by: ORDS_PRO         (procedure in CREATEOR.PRG)
    Called by: ORDT_PRO         (procedure in EDITORDE.PRG)
Contains: ENDORDR               (Params: none)
    Called by: ORDS_PRO         (procedure in CREATEOR.PRG)
    Called by: ORDT_PRO         (procedure in EDITORDE.PRG)
Contains: STARTOR               (Params: none)
    Called by: CREATEOR.PRG
    Called by: EDITORDE.PRG
    Called by: ORDS_PRO         (procedure in CREATEOR.PRG)
    Called by: ORDT_PRO         (procedure in EDITORDE.PRG)
        Calls: STARTHLP.PRG
Contains: DISPORD               (Params: none)
    Called by: CREATEO          (procedure in CREATEOR.PRG)
    Called by: GETORDER         (procedure in EDITORDE.PRG)
Contains: DISPORD2              (Params: none)
    Called by: VIEWORDE         (procedure in CREATEOR.PRG)
    Called by: ORDT_PRO         (procedure in EDITORDE.PRG)
    Called by: GETORDER         (procedure in EDITORDE.PRG)
Contains: TOMP_PRO              (Params: none)
    Called by: CREATEOR.PRG
    Called by: EDITORDE.PRG
        Calls: 1PROALL          (procedure in CREATEOR.PRG)
        Calls: 1PROONE          (procedure in CREATEOR.PRG)
        Calls: 1PRO1PRI         (procedure in CREATEOR.PRG)
        Calls: PHASEBY          (procedure in CREATEOR.PRG)
        Calls: VIEWTOP          (procedure in CREATEOR.PRG)
Contains: 1PROALL               (Params: none)
    Called by: TOMP_PRO         (procedure in CREATEOR.PRG)
        Calls: FINDPRO          (procedure in CREATEOR.PRG)
Contains: 1PROONE               (Params: none)
    Called by: TOMP_PRO         (procedure in CREATEOR.PRG)
        Calls: SETBROWS.PRG
        Calls: ENDBROWS.PRG
        Calls: FINDPRO          (procedure in CREATEOR.PRG)
Contains: 1PRO1PRI              (Params: none)
    Called by: TOMP_PRO         (procedure in CREATEOR.PRG)
        Calls: FINDPRO          (procedure in CREATEOR.PRG)
Contains: PHASEBY               (Params: none)
    Called by: TOMP_PRO         (procedure in CREATEOR.PRG)
        Calls: VIEWTOP          (procedure in CREATEOR.PRG)
        Calls: FINDPRO          (procedure in CREATEOR.PRG)
Contains: VIEWTOP               (Params: none)
```

414

```
       Called by: TOMP_PRO              (procedure in CREATEOR:PRG)
       Called by: PHASEBY               (procedure in CREATEOR.PRG)
           Calls: SETBROWS.PRG
           Calls: ENDBROWS.PRG
     Contains: DISPPROF                 (Params: none)
       Called by: NEWPROFI              (procedure in CREATEPR.PRG)
       Called by: VIEWPROF              (procedure in CREATEPR.PRG)
       Called by: SELEPROF              (procedure in EDITPROF.PRG)
       Called by: CHANGEAP              (procedure in EDITPROF.PRG)
       Called by: CHANGEAC              (procedure in EDITPROF.PRG)
       Called by: CHANGEI               (procedure in EDITPROF.PRG)
       Called by: CHANGEIF              (procedure in EDITPROF.PRG)
       Called by: CHANGECI              (procedure in EDITPROF.PRG)
       Called by: CHANGEH2              (procedure in EDITPROF.PRG)
       Called by: CHANGEME              (procedure in EDITPROF.PRG)
       Called by: SELECTPRO             (procedure in DELPROF.PRG)
       Called by: FINDPRO               (procedure in CREATEOR.PRG)
     Contains: FINDPRO                  (Params: none)
       Called by: 1PROALL               (procedure in CREATEOR.PRG)
       Called by: 1PROONE               (procedure in CREATEOR.PRG)
       Called by: 1PRO1PRI              (procedure in CREATEOR.PRG)
       Called by: PHASEBY               (procedure in CREATEOR.PRG)
           Calls: SETBROWS.PRG
           Calls: ENDBROWS.PRG
           Calls: DISPPROF              (procedure in CREATEPR.PRG)
     Contains: DISPTAS                  (Params: none)
     Contains: FINDTA                   (Params: none)
       Called by: EDITTASK.PRG
       Called by: COPT_PRO              (procedure in CREATETC.PRG)
       Called by: EDIT_PRO              (procedure in EDITTASK.PRG)
       Called by: DELT_PRO              (procedure in DELTASK.PRG)
           Calls: SETBROWS.PRG
           Calls: ENDBROWS.PRG
           Calls: DISPTASK              (procedure in CREATETA.PRG)


-----------------------------------------------------------------------
EDITORDE.PRG -- Last updated:   04/12/94 at 21:47

     Contains: ORDT_PRO                 (Params: none)
       Called by: EDITORDE.PRG
           Calls: STARTOR               (procedure in CREATEOR.PRG)
           Calls: GETORDER              (procedure in EDITORDE.PRG)
           Calls: VIEWORDE              (procedure in CREATEOR.PRG)
           Calls: DISPORD2              (procedure in CREATEOR.PRG)
           Calls: ASSIGNTA              (procedure in CREATEOR.PRG)
           Calls: EDITPLEN              (procedure in EDITORDE.PRG)
           Calls: ENDORDR               (procedure in CREATEOR.PRG)
     Contains: GETORDER                 (Params: none)
       Called by: ORDT_PRO              (procedure in EDITORDE.PRG)
       Called by: DORD_PRO              (procedure in DELORDE.PRG)
       Called by: ORDR_PRO              (procedure in ORDEREP1.PRG)
           Calls: SETBROWS.PRG
           Calls: ENDBROWS.PRG
           Calls: DISPORD2              (procedure in CREATEOR.PRG)
           Calls: VIEWORDE              (procedure in CREATEOR.PRG)
           Calls: DISPORD               (procedure in CREATEOR.PRG)
           Calls: DISPORD3              (procedure in ORDEREP1.PRG)
           Calls: CALCPOL.PRG
```

```
        Calls: CALCAMMO.PRG
       Calls: CALCOTHE.PRG
Contains: VIEWORDE                  (Params: none)
   Called by: ORDS_PRO              (procedure in CREATEOR.PRG)
   Called by: ORDT_PRO              (procedure in EDITORDE.PRG)
   Called by: GETORDER              (procedure in EDITORDE.PRG)
        Calls: SETBROWS.PRG
        Calls: ENDBROWS.PRG
        Calls: DISPORD2             (procedure in CREATEOR.PRG)
        Calls: DISPORD3             (procedure in ORDEREP1.PRG)
Contains: ASSIGNTA                  (Params: none)
   Called by: ORDS_PRO              (procedure in CREATEOR.PRG)
   Called by: ORDT_PRO              (procedure in EDITORDE.PRG)
Contains: ENDORDR                   (Params: none)
   Called by: ORDS_PRO              (procedure in CREATEOR.PRG)
   Called by: ORDT_PRO              (procedure in EDITORDE.PRG)
Contains: STARTOR                   (Params: none)
   Called by: CREATEOR.PRG
   Called by: EDITORDE.PRG
   Called by: ORDS_PRO              (procedure in CREATEOR.PRG)
   Called by: ORDT_PRO              (procedure in EDITORDE.PRG)
        Calls: STARTHLP.PRG
Contains: DISPORD                   (Params: none)
   Called by: CREATEO               (procedure in CREATEOR.PRG)
   Called by: GETORDER              (procedure in EDITORDE.PRG)
Contains: DISPORD2                  (Params: none)
   Called by: VIEWORDE              (procedure in CREATEOR.PRG)
   Called by: ORDT_PRO              (procedure in EDITORDE.PRG)
   Called by: GETORDER              (procedure in EDITORDE.PRG)
Contains: EDITPLEN                  (Params: none)
   Called by: ORDT_PRO              (procedure in EDITORDE.PRG)
Contains: TOMP_PRO                  (Params: none)
   Called by: CREATEOR.PRG
   Called by: EDITORDE.PRG
        Calls: 1PROALL             (procedure in CREATEOR.PRG)
        Calls: 1PROONE             (procedure in CREATEOR.PRG)
        Calls: 1PRO1PRI            (procedure in CREATEOR.PRG)
        Calls: PHASEBY             (procedure in CREATEOR.PRG)
        Calls: VIEWTOP             (procedure in CREATEOR.PRG)
Contains: 1PROALL                   (Params: none)
   Called by: TOMP_PRO             (procedure in CREATEOR.PRG)
        Calls: FINDPRO             (procedure in CREATEOR.PRG)
Contains: 1PROONE                   (Params: none)
   Called by: TOMP_PRO             (procedure in CREATEOR.PRG)
        Calls: SETBROWS.PRG
        Calls: ENDBROWS.PRG
        Calls: FINDPRO             (procedure in CREATEOR.PRG)
Contains: 1PRO1PRI                  (Params: none)
   Called by: TOMP_PRO             (procedure in CREATEOR.PRG)
        Calls: FINDPRO             (procedure in CREATEOR.PRG)
Contains: PHASEBY                   (Params: none)
   Called by: TOMP_PRO             (procedure in CREATEOR.PRG)
        Calls: VIEWTOP             (procedure in CREATEOR.PRG)
        Calls: FINDPRO             (procedure in CREATEOR.PRG)
Contains: VIEWTOP                   (Params: none)
   Called by: TOMP_PRO             (procedure in CREATEOR.PRG)
   Called by: PHASEBY             (procedure in CREATEOR.PRG)
        Calls: SETBROWS.PRG
        Calls: ENDBROWS.PRG
```

416

```
Contains: DISPPROF              (Params: none)
   Called by: NEWPROFI           (procedure in CREATEPR.PRG)
   Called by: VIEWPROF           (procedure in CREATEPR.PRG)
   Called by: SELEPROF           (procedure in EDITPROF.PRG)
   Called by: CHANGEAP           (procedure in EDITPROF.PRG)
   Called by: CHANGEAC           (procedure in EDITPROF.PRG)
   Called by: CHANGEI            (procedure in EDITPROF.PRG)
   Called by: CHANGEIF           (procedure in EDITPROF.PRG)
   Called by: CHANGECI           (procedure in EDITPROF.PRG)
   Called by: CHANGEH2           (procedure in EDITPROF.PRG)
   Called by: CHANGEME           (procedure in EDITPROF.PRG)
   Called by: SELECTPRO          (procedure in DELPROF.PRG)
   Called by: FINDPRO            (procedure in CREATEOR.PRG)
Contains: FINDPRO               (Params: none)
   Called by: 1PROALL           (procedure in CREATEOR.PRG)
   Called by: 1PROONE           (procedure in CREATEOR.PRG)
   Called by: 1PRO1PRI          (procedure in CREATEOR.PRG)
   Called by: PHASEBY           (procedure in CREATEOR.PRG)
      Calls: SETBROWS.PRG
      Calls: ENDBROWS.PRG
      Calls: DISPPROF           (procedure in CREATEPR.PRG)
Contains: DISPTAS               (Params: none)
Contains: FINDTA                (Params: none)
   Called by: EDITTASK.PRG
   Called by: COPT_PRO           (procedure in CREATETC.PRG)
   Called by: EDIT_PRO           (procedure in EDITTASK.PRG)
   Called by: DELT_PRO           (procedure in DELTASK.PRG)
      Calls: SETBROWS.PRG
      Calls: ENDBROWS.PRG
      Calls: DISPTASK           (procedure in CREATETA.PRG)
Contains: STRE_PRO              (Params: none)
   Called by: EDITORDE.PRG
      Calls: 1STRALL            (procedure in EDITORDE.PRG)
      Calls: 1STRONE            (procedure in EDITORDE.PRG)
      Calls: 1STR1PRI           (procedure in EDITORDE.PRG)
      Calls: PHASEBYS           (procedure in EDITORDE.PRG)
      Calls: VIEWSTR            (procedure in EDITORDE.PRG)
Contains: 1STRALL               (Params: none)
   Called by: STRE_PRO           (procedure in EDITORDE.PRG)
Contains: 1STRONE               (Params: none)
   Called by: STRE_PRO           (procedure in EDITORDE.PRG)
      Calls: SETBROWS.PRG
      Calls: ENDBROWS.PRG
Contains: 1STR1PRI              (Params: none)
   Called by: STRE_PRO           (procedure in EDITORDE.PRG)
Contains: PHASEBYS              (Params: none)
   Called by: STRE_PRO           (procedure in EDITORDE.PRG)
      Calls: SETBROW2.PRG
      Calls: ENDBROWS.PRG
Contains: VIEWSTR               (Params: none)
   Called by: STRE_PRO           (procedure in EDITORDE.PRG)
      Calls: SETBROWS.PRG
      Calls: ENDBROWS.PRG
```

DELORDE.PRG -- Last updated: 04/10/94 at 19:45

```
   Contains: DORD_PRO            (Params: none)
```

```
          Called by: DELORDE.PRG
               Calls: GETORDER              (procedure in EDITORDE.PRG)
        Contains: GETORDER                  (Params: none)
          Called by: ORDT_PRO               (procedure in EDITORDE.PRG)
          Called by: DORD_PRO               (procedure in DELORDE.PRG)

          Called by: ORDR_PRO               (procedure in ORDEREP1.PRG)
               Calls: SETBROWS.PRG
               Calls: ENDBROWS.PRG
               Calls: DISPORD2              (procedure in CREATEOR.PRG)
               Calls: VIEWORDE              (procedure in CREATEOR.PRG)
               Calls: DISPORD               (procedure in CREATEOR.PRG)
               Calls: DISPORD3              (procedure in ORDEREP1.PRG)
               Calls: CALCPOL.PRG
               Calls: CALCAMMO.PRG
               Calls: CALCOTHE.PRG
        Contains: DISPORD                   (Params: none)
          Called by: CREATEO                (procedure in CREATEOR.PRG)
          Called by: GETORDER               (procedure in EDITORDE.PRG)
        Contains: DISPORD2                  (Params: none)
          Called by: VIEWORDE               (procedure in CREATEOR.PRG)
          Called by: ORDT_PRO               (procedure in EDITORDE.PRG)
          Called by: GETORDER               (procedure in EDITORDE.PRG)
        Contains: VIEWORDE                  (Params: none)
          Called by: ORDS_PRO               (procedure in CREATEOR.PRG)
          Called by: ORDT_PRO               (procedure in EDITORDE.PRG)
          Called by: GETORDER               (procedure in EDITORDE.PRG)
               Calls: SETBROWS.PRG
               Calls: ENDBROWS.PRG
               Calls: DISPORD2              (procedure in CREATEOR.PRG)
               Calls: DISPORD3              (procedure in ORDEREP1.PRG)


----------------------------------------------------------------
INSTREPT.PRG -- Last updated:  04/10/94 at 19:45

     Contains: ORDR_BAR1                    (Params: none)
     Contains: ORDR_BAR2                    (Params: none)
     Contains: ORDR_BAR3                    (Params: none)
     Contains: ORDR_BAR4                    (Params: none)
     Contains: ORDR_BAR5                    (Params: none)


----------------------------------------------------------------
ORDEREP1.PRG -- Last updated:  04/12/94 at 21:49

     Contains: ORDR_PRO                     (Params: none)
        Called by: ORDEREP1.PRG
               Calls: STARTORD              (procedure in ORDEREP1.PRG)
               Calls: OLDORD                (procedure in ORDEREP1.PRG)
               Calls: GETORDER              (procedure in EDITORDE.PRG)
               Calls: DISPORD3              (procedure in ORDEREP1.PRG)
               Calls: PRINTPOL.PRG
               Calls: PRINTAMM.PRG
               Calls: PRINTOTH.PRG
               Calls: ORDREND               (procedure in ORDEREP1.PRG)
        Contains: OLDORD                    (Params: none)
          Called by: ORDR_PRO               (procedure in ORDEREP1.PRG)
        Contains: GETORDER                  (Params: none)
          Called by: ORDT_PRO               (procedure in EDITORDE.PRG)
                              418
```

```
     Called by: DORD_PRO          (procedure in DELORDE.PRG)
     Called by: ORDR_PRO          (procedure in ORDEREP1.PRG)
         Calls: SETBROWS.PRG
         Calls: ENDBROWS.PRG
         Calls: DISPORD2           (procedure in CREATEOR.PRG)
         Calls: VIEWORDE           (procedure in CREATEOR.PRG)
         Calls: DISPORD            (procedure in CREATEOR.PRG)
         Calls: DISPORD3           (procedure in ORDEREP1.PRG)
         Calls: CALCPOL.PRG
         Calls: CALCAMMO.PRG
         Calls: CALCOTHE.PRG
   Contains: VIEWORDE             (Params: none)
     Called by: ORDS_PRO          (procedure in CREATEOR.PRG)
     Called by: ORDT_PRO          (procedure in EDITORDE.PRG)
     Called by: GETORDER          (procedure in EDITORDE.PRG)
         Calls: SETBROWS.PRG
         Calls: ENDBROWS.PRG
         Calls: DISPORD2           (procedure in CREATEOR.PRG)
         Calls: DISPORD3           (procedure in ORDEREP1.PRG)
   Contains: DISPORD3             (Params: none)
     Called by: VIEWORDE          (procedure in CREATEOR.PRG)
     Called by: GETORDER          (procedure in EDITORDE.PRG)
     Called by: ORDR_PRO          (procedure in ORDEREP1.PRG)
   Contains: STARTORD             (Params: none)
     Called by: ORDR_PRO          (procedure in ORDEREP1.PRG)
   Contains: ORDREND              (Params: none)
     Called by: ORDR_PRO          (procedure in ORDEREP1.PRG)


-----------------------------------------------------------------
INSTSYSR.PRG -- Last updated:  04/10/94 at 19:45

   Contains: SYSR_BAR1            (Params: none)
   Contains: SYSR_BAR2            (Params: none)
   Contains: SYSR_BAR3            (Params: none)
   Contains: SYSR_BAR4            (Params: none)
   Contains: SYSR_BAR5            (Params: none)
   Contains: SYSR_BAR6            (Params: none)
   Contains: SYSR_BAR7            (Params: none)
   Contains: SYSR_BAR8            (Params: none)
   Contains: SYSR_BAR9            (Params: none)
   Contains: SYSR_BA10            (Params: none)
   Contains: SYSR_BA11            (Params: none)
   Contains: SYSR_BA12            (Params: none)


-----------------------------------------------------------------
SYSREP.PRG -- Last updated:  04/10/94 at 19:45

   Contains: SYSR_PRO             (Params: none)
     Called by: SYSREP.PRG
         Calls: SRCRPT            (procedure in SYSREP.PRG)
         Calls: SRCLINR           (procedure in SYSREP.PRG)
         Calls: UNITRPT           (procedure in SYSREP.PRG)
         Calls: UNITLINR          (procedure in SYSREP.PRG)
         Calls: PROFILER          (procedure in SYSREP.PRG)
         Calls: TASKORGR          (procedure in SYSREP.PRG)
         Calls: TASKUNIT          (procedure in SYSREP.PRG)
         Calls: ORDERRPT          (procedure in SYSREP.PRG)
```

```
          Calls: ORDERTAS           (procedure in SYSREP.PRG)
          Calls: ORDTASUN           (procedure in SYSREP.PRG)
          Calls: PRNSETUP.PRG
          Calls: SYSREND            (procedure in SYSREP.PRG)
  Contains: SRCRPT                  (Params: none)
     Called by: SYSR_PRO            (procedure in SYSREP.PRG)
          Calls: SRCLST             (procedure in SYSREP.PRG)
  Contains: SRCLST                  (Params: none)
     Called by: SRCRPT              (procedure in SYSREP.PRG)
  Contains: SRCLINR                 (Params: none)
     Called by: SYSR_PRO            (procedure in SYSREP.PRG)
          Calls: GETSRC             (procedure in SYSREP.PRG)
          Calls: LINLST             (procedure in SYSREP.PRG)
  Contains: LINLST                  (Params: none)
     Called by: SRCLINR             (procedure in SYSREP.PRG)
  Contains: UNITRPT                 (Params: none)
     Called by: SYSR_PRO            (procedure in SYSREP.PRG)
          Calls: UNITLST            (procedure in SYSREP.PRG)
  Contains: UNITLINR                (Params: none)
     Called by: SYSR_PRO            (procedure in SYSREP.PRG)
          Calls: UNITLIN            (procedure in SYSREP.PRG)
  Contains: PROFILER                (Params: none)
     Called by: SYSR_PRO            (procedure in SYSREP.PRG)
          Calls: PROFLST            (procedure in SYSREP.PRG)
  Contains: TASKORGR                (Params: none)
     Called by: SYSR_PRO            (procedure in SYSREP.PRG)
          Calls: TASKLST            (procedure in SYSREP.PRG)
  Contains: TASKUNIT                (Params: none)
     Called by: SYSR_PRO            (procedure in SYSREP.PRG)
          Calls: TASKUNI            (procedure in SYSREP.PRG)
  Contains: ORDERRPT                (Params: none)
     Called by: SYSR_PRO            (procedure in SYSREP.PRG)
          Calls: ORDERLST           (procedure in SYSREP.PRG)
  Contains: ORDERTAS                (Params: none)
     Called by: SYSR_PRO            (procedure in SYSREP.PRG)
          Calls: ORDERTP            (procedure in SYSREP.PRG)
  Contains: ORDTASUN                (Params: none)
     Called by: SYSR_PRO            (procedure in SYSREP.PRG)
          Calls: ORDERTUP           (procedure in SYSREP.PRG)
  Contains: SYSREND                 (Params: none)
     Called by: SYSR_PRO            (procedure in SYSREP.PRG)
  Contains: STARTSRP                (Params: none)
     Called by: SYSREP.PRG
          Calls: STARTHLP.PRG
  Contains: GETSRC                  (Params: none)
     Called by: SRCLINR             (procedure in SYSREP.PRG)
          Calls: SETBROWS.PRG
          Calls: ENDBROWS.PRG
          Calls: DISPSRC            (procedure in SYSREP.PRG)
  Contains: DISPSRC                 (Params: none)
     Called by: GETSRC              (procedure in SYSREP.PRG)
  Contains: DISPPROF                (Params: none)
     Called by: NEWPROFI            (procedure in CREATEPR.PRG)
     Called by: VIEWPROF            (procedure in CREATEPR.PRG)
     Called by: SELEPROF            (procedure in EDITPROF.PRG)
     Called by: CHANGEAP            (procedure in EDITPROF.PRG)
     Called by: CHANGEAC            (procedure in EDITPROF.PRG)
     Called by: CHANGEI             (procedure in EDITPROF.PRG)
     Called by: CHANGEIF            (procedure in EDITPROF.PRG)
```

420

```
       Called by: CHANGECI            (procedure in EDITPROF.PRG)
       Called by: CHANGEH2            (procedure in EDITPROF.PRG)
       Called by: CHANGEME            (procedure in EDITPROF.PRG)
       Called by: SELECTPRO           (procedure in DELPROF.PRG)
       Called by: FINDPRO             (procedure in CREATEOR.PRG)
   Contains: LISTSRC                  (Params: none)
   Contains: LISTPROF                 (Params: none)
   Contains: PRIN_PRO                 (Params: none)
       Called by: SYSREP.PRG
       Called by: PRINTPOL.PRG
       Called by: PRINTAMM.PRG
       Called by: PRINTOTH.PRG
           Calls: PRNSETUP.PRG
           Calls: STARTPLR            (procedure in PRINTPOL.PRG)
           Calls: REPORTHD            (procedure in PRINTPOL.PRG)
           Calls: TFHEAD              (procedure in PRINTPOL.PRG)
           Calls: LINPOL              (procedure in PRINTPOL.PRG)
           Calls: POLCALC             (procedure in PRINTPOL.PRG)
           Calls: PUNITPOL            (procedure in PRINTPOL.PRG)
           Calls: PTASKPOL            (procedure in PRINTPOL.PRG)
           Calls: PORDEPOL            (procedure in PRINTPOL.PRG)
           Calls: LINAMM              (procedure in PRINTAMM.PRG)
           Calls: AMMOCALC            (procedure in PRINTAMM.PRG)
           Calls: PUNITAMM            (procedure in PRINTAMM.PRG)
           Calls: PTASKAMM            (procedure in PRINTAMM.PRG)
           Calls: PORDEAMM            (procedure in PRINTAMM.PRG)
           Calls: OTHECALC            (procedure in PRINTOTH.PRG)
           Calls: PUNITOTH            (procedure in PRINTOTH.PRG)
           Calls: PTASKOTH            (procedure in PRINTOTH.PRG)
           Calls: PORDEOTH            (procedure in PRINTOTH.PRG)
   Contains: SRCLST                   (Params: none)
       Called by: SRCRPT             (procedure in SYSREP.PRG)
   Contains: UNITLST                  (Params: none)
       Called by: UNITRPT            (procedure in SYSREP.PRG)
   Contains: UNITLIN                  (Params: none)
       Called by: UNITLINR           (procedure in SYSREP.PRG)
   Contains: PROFLST                  (Params: none)
       Called by: PROFILER           (procedure in SYSREP.PRG)
   Contains: TASKLST                  (Params: none)
       Called by: TASKORGR           (procedure in SYSREP.PRG)
   Contains: TASKUNI                  (Params: none)
       Called by: TASKUNIT           (procedure in SYSREP.PRG)
   Contains: ORDERTP                  (Params: none)
       Called by: ORDERTAS           (procedure in SYSREP.PRG)
   Contains: ORDERLST                 (Params: none)
       Called by: ORDERRPT           (procedure in SYSREP.PRG)
   Contains: ORDERTUP                 (Params: none)
       Called by: ORDTASUN           (procedure in SYSREP.PRG)


---------------------------------------------------------------
RESTOREF.PRG -- Last updated:  04/10/94 at 19:45

   Contains: RESTORE                  (Params: none)
       Called by: RESTOREF.PRG
           Calls: GETSTATU            (procedure in RESTOREF.PRG)
   Contains: GETSTATU                 (Params: none)
       Called by: RESTORE            (procedure in RESTOREF.PRG)
```

```
-------------------------------------------------------------------
INSTPHLP.PRG -- Last updated:  04/10/94 at 19:45

    Contains: PHEL_BAR1              (Params: none)
    Contains: PHEL_BAR2              (Params: none)
    Contains: PHEL_BAR3              (Params: ncne)


-------------------------------------------------------------------
PRNTHELP.PRG -- Last updated:  04/10/94 at 19:45

    Contains: PHEL_PRO               (Params: none)
       Called by: PRNTHELP.PRG
            Calls: PRNONEH           (procedure in PRNTHELP.PRG)
            Calls: PRNALLH           (procedure in PRNTHELP.PRG)
            Calls: ENDPHEL           (procedure in PRNTHELP.PRG)
    Contains: PRNONEH                (Params: none)
       Called by: PHEL_PRO           (procedure in PRNTHELP.PRG)
            Calls: BROWHELP          (procedure in PRNTHELP.PRG)
            Calls: PRNTMEMO.PRG
    Contains: PRNALLH                (Params: none)
       Called by: PHEL_PRO           (procedure in PRNTHELP.PRG)
            Calls: PRNTMEMO.PRG
    Contains: ENDPHEL                (Params: none)
       Called by: PHEL_PRO           (procedure in PRNTHELP.PRG)
    Contains: BROWHELP               (Params: none)
       Called by: PRNONEH            (procedure in PRNTHELP.PRG)
            Calls: SETBROWS.PRG
            Calls: ENDBROWS.PRG


-------------------------------------------------------------------
PRINTPOL.PRG -- Last updated:  04/13/94 at  5:53

    Contains: PRIN_PRO               (Params: none)
       Called by: SYSREP.PRG
       Called by: PRINTPOL.PRG
       Called by: PRINTAMM.PRG
       Called by: PRINTOTH.PRG
            Calls: PRNSETUP.PRG
            Calls: STARTPLR          (procedure in PRINTPOL.PRG)
            Calls: REPORTHD          (procedure in PRINTPOL.PRG)
            Calls: TFHEAD            (procedure in PRINTPOL.PRG)
            Calls: LINPOL            (procedure in PRINTPOL.PRG)
            Calls: POLCALC           (procedure in PRINTPOL.PRG)
            Calls: PUNITPOL          (procedure in PRINTPOL.PRG)
            Calls: PTASKPOL          (procedure in PRINTPOL.PRG)
            Calls: PORDEPOL          (procedure in PRINTPOL.PRG)
            Calls: LINAMM            (procedure in PRINTAMM.PRG)
            Calls: AMMOCALC          (procedure in PRINTAMM.PRG)
            Calls: PUNITAMM          (procedure in PRINTAMM.PRG)
            Calls: PTASKAMM          (procedure in PRINTAMM.PRG)
            Calls: PORDEAMM          (procedure in PRINTAMM.PRG)
            Calls: OTHECALC          (procedure in PRINTOTH.PRG)
            Calls: PUNITOTH          (procedure in PRINTOTH.PRG)
            Calls: PTASKOTH          (procedure in PRINTOTH.PRG)
            Calls: PORDEOTH          (procedure in PRINTOTH.PRG)
    Contains: STARTPLR               (Params: none)
```

```
          Called by: PRINTPOL.PRG
          Called by: PRINTAMM.PRG
          Called by: PRINTOTH.PRG
          Called by: PRIN_PRO              (procedure in SYSREP.PRG)
             Calls: STARTHLP.PRG
       Contains: POLCALC                   (Params: none)
          Called by: CALCPOL.PRG
          Called by: PRIN_PRO              (procedure in SYSREP.PRG)
       Contains: PUNITPOL                  (Params: none)
          Called by: PRIN_PRO              (procedure in SYSREP.PRG)
       Contains: PTASKPOL                  (Params: none)
          Called by: PRIN_PRO              (procedure in SYSREP.PRG)
       Contains: PORDEPOL                  (Params: none)
          Called by: PRIN_PRO              (procedure in SYSREP.PRG)
       Contains: LINPOL                    (Params: none)
          Called by: PRIN_PRO              (procedure in SYSREP.PRG)
       Contains: REPORTHD                  (Params: none)
          Called by: PRIN_PRO              (procedure in SYSREP.PRG)
       Contains: TFHEAD                    (Params: none)
          Called by: PRIN_PRO              (procedure in SYSREP.PRG)
       Contains: DETA_PRO                  (Params: none)
          Called by: PRINTPOL.PRG
          Called by: PRINTAMM.PRG
          Called by: PRINTOTH.PRG
       Contains: DETA_BAR1                 (Params: none)
       Contains: DETA_BAR2                 (Params: none)
       Contains: DETA_BAR3                 (Params: none)
       Contains: DETA_BAR4                 (Params: none)
       Contains: PRIN_BAR1                 (Params: none)
       Contains: PRIN_BAR2                 (Params: none)
       Contains: PRIN_BAR3                 (Params: none)
       Contains: PRIN_BAR4                 (Params: none)
       Contains: PRIN_BAR5                 (Params: none)


------------------------------------------------------------------

PRINTAMM.PRG -- Last updated:  04/10/94 at 19:45

       Contains: PRIN_PRO                  (Params: none)
          Called by: SYSREP.PRG
          Called by: PRINTPOL.PRG
          Called by: PRINTAMM.PRG
          Called by: PRINTOTH.PRG
             Calls: PRNSETUP.PRG
             Calls: STARTPLR               (procedure in PRINTPOL.PRG)
             Calls: REPORTHD               (procedure in PRINTPOL.PRG)
             Calls: TFHEAD                 (procedure in PRINTPOL.PRG)
             Calls: LINPOL                 (procedure in PRINTPOL.PRG)
             Calls: POLCALC                (procedure in PRINTPOL.PRG)
             Calls: PUNITPOL               (procedure in PRINTPOL.PRG)
             Calls: PTASKPOL               (procedure in PRINTPOL.PRG)
             Calls: PORDEPOL               (procedure in PRINTPOL.PRG)
             Calls: LINAMM                 (procedure in PRINTAMM.PRG)
             Calls: AMMOCALC               (procedure in PRINTAMM.PRG)
             Calls: PUNITAMM               (procedure in PRINTAMM.PRG)
             Calls: PTASKAMM               (procedure in PRINTAMM.PRG)
             Calls: PORDEAMM               (procedure in PRINTAMM.PRG)
             Calls: OTHECALC               (procedure in PRINTOTH.PRG)
             Calls: PUNITOTH               (procedure in PRINTOTH.PRG)
```

```
              Calls: PTASKOTH              (procedure in PRINTOTH.PRG)
              Calls: PORDEOTH              (procedure in PRINTOTH.PRG)
          Contains: STARTPLR               (Params: none)
              Called by: PRINTPOL.PRG
              Called by: PRINTAMM.PRG
              Called by: PRINTOTH.PRG
              Called by: PRIN_PRO          (procedure in SYSREP.PRG)
                  Calls: STARTHLP.PRG
          Contains: AMMOCALC               (Params: none)
              Called by: CALCAMMO.PRG
              Called by: PRIN_PRO          (procedure in SYSREP.PRG)
          Contains: PUNITAMM               (Params: none)
              Called by: PRIN_PRO          (procedure in SYSREP.PRG)
          Contains: PTASKAMM               (Params: none)
              Called by: PRIN_PRO          (procedure in SYSREP.PRG)
          Contains: PORDEAMM               (Params: none)
              Called by: PRIN_PRO          (procedure in SYSREP.PRG)
          Contains: LINAMM                 (Params: none)
              Called by: PRIN_PRO          (procedure in SYSREP.PRG)
          Contains: REPORTHD               (Params: none)
              Called by: PRIN_PRO          (procedure in SYSREP.PRG)
          Contains: DETA_PRO               (Params: none)
              Called by: PRINTPOL.PRG
              Called by: PRINTAMM.PRG
              Called by: PRINTOTH.PRG
          Contains: DETA_BAR1              (Params: none)
          Contains: DETA_BAR2              (Params: none)
          Contains: DETA_BAR3              (Params: none)
          Contains: DETA_BAR4              (Params: none)
          Contains: PRIN_BAR1              (Params: none)
          Contains: PRIN_BAR2              (Params: none)
          Contains: PRIN_BAR3              (Params: none)
          Contains: PRIN_BAR4              (Params: none)
          Contains: PRIN_BAR5              (Params: none)


--------------------------------------------------------------------

PRINTOTH.PRG -- Last updated:  04/10/94 at 19:45

          Contains: PRIN_PRO               (Params: none)
              Called by: SYSREP.PRG
              Called by: PRINTPOL.PRG
              Called by: PRINTAMM.PRG
              Called by: PRINTOTH.PRG
                  Calls: PRNSETUP.PRG
                  Calls: STARTPLR          (procedure in PRINTPOL.PRG)
                  Calls: REPORTHD          (procedure in PRINTPOL.PRG)
                  Calls: TFHEAD            (procedure in PRINTPOL.PRG)
                  Calls: LINPOL            (procedure in PRINTPOL.PRG)
                  Calls: POLCALC           (procedure in PRINTPOL.PRG)
                  Calls: PUNITPOL          (procedure in PRINTPOL.PRG)
                  Calls: PTASKPOL          (procedure in PRINTPOL.PRG)
                  Calls: PORDEPOL          (procedure in PRINTPOL.PRG)
                  Calls: LINAMM            (procedure in PRINTAMM.PRG)
                  Calls: AMMOCALC          (procedure in PRINTAMM.PRG)
                  Calls: PUNITAMM          (procedure in PRINTAMM.PRG)
                  Calls: PTASKAMM          (procedure in PRINTAMM.PRG)
                  Calls: PORDEAMM          (procedure in PRINTAMM.PRG)
                  Calls: OTHECALC          (procedure in PRINTOTH.PRG)
```

```
            Calls: PUNITOTH          (procedure in PRINTOTH.PRG)
            Calls: PTASKOTH          (procedure in PRINTOTH.PRG)
            Calls: PORDEOTH          (procedure in PRINTOTH.PRG)
      Contains: STARTPLR             (Params: none)
         Called by: PRINTPOL.PRG
         Called by: PRINTAMM.PRG
         Called by: PRINTOTH.PRG
         Called by: PRIN_PRO         (procedure in SYSREP.PRG)
            Calls: STARTHLP.PRG
      Contains: OTHECALC             (Params: none)
         Called by: PRIN_PRO         (procedure in SYSREP.PRG)
      Contains: PUNITOTH             (Params: none)
         Called by: PRIN_PRO         (procedure in SYSREP.PRG)
      Contains: PTASKOTH             (Params: none)
         Called by: PRIN_PRO         (procedure in SYSREP.PRG)
      Contains: PORDEOTH             (Params: none)
         Called by: PRIN_PRO         (procedure in SYSREP.PRG)
      Contains: REPORTHD             (Params: none)
         Called by: PRIN_PRO         (procedure in SYSREP.PRG)
      Contains: DETA_PRO             (Params: none)
         Called by: PRINTPOL.PRG
         Called by: PRINTAMM.PRG
         Called by: PRINTOTH.PRG
      Contains: DETA_BAR1            (Params: none)
      Contains: DETA_BAR2            (Params: none)
      Contains: DETA_BAR3            (Params: none)
      Contains: PRIN_BAR1            (Params: none)
      Contains: PRIN_BAR2            (Params: none)
      Contains: PRIN_BAR3            (Params: none)
      Contains: PRIN_BAR4            (Params: none)
      Contains: PRIN_BAR5            (Params: none)


------------------------------------------------------------------
CALCPOL.PRG -- Last updated:  04/10/94 at 19:45

      Contains: POLCALC              (Params: none)
         Called by: CALCPOL.PRG
         Called by: PRIN_PRO         (procedure in SYSREP.PRG)


------------------------------------------------------------------
CALCAMMO.PRG -- Last updated:  04/10/94 at 19:45

      Contains: AMMOCALC             (Params: none)
         Called by: CALCAMMO.PRG
         Called by: PRIN_PRO         (procedure in SYSREP.PRG)


------------------------------------------------------------------
CALCOTHE.PRG -- Last updated:  04/12/94 at 21:55

      Contains: OTHERCAL             (Params: none)
         Called by: CALCOTHE.PRG
```

----

34 index files in the system
    PROGRAM
    SUBJECT
    SRC
    UNIT_DESIG
    STRENGTH
    UNITNAME
    ECHELON
    LIN
    UNIQUELI
    DODIC
    POL_TYPE
    TASKORGNAM
    UNIQUEUN
    UNIQUETA
    NUMBER
    TASKPHAS
    PROFILENAM
    TASKPHASE
    UNIQUENU
    PHASE
    UNIQUETN
    PHASETAS
    MED_NAME
    WATER_NAME
    RATION
    PROFILE
    INTENSITY
    PROFILEDES
    POSTURE
    CDE
    NAME
    DESCRIPTIO
    PARENT_UNI
    UNITDODI

PROGRAM is an index tag

This index file is associated with database(s):
      : HELP.DBF

Used by: REINDX.PRG

SUBJECT is an index tag

This index file is associated with database(s):
      : HELP.DBF

Used by: REINDX.PRG

SRC is an index tag

This index file is associated with database(s):
        : CSRCLIST.DBF
        : SRCLIST.DBF
        : SRCLINQT.DBF
        : CSRCLINQ.DBF
        : TASKORG.DBF

Used by: REINDX.PRG

UNIT_DESIG is an index tag

This index file is associated with database(s):
        : CSRCLIST.DBF
        : SRCLIST.DBF

Used by: REINDX.PRG

STRENGTH is an index tag

This index file is associated with database(s):
        : CSRCLIST.DBF

Used by: REINDX.PRG

UNITNAME is an index tag

This index file is associated with database(s):
        : CSRCLIST.DBF
        : CSRCLINQ.DBF
        : TASKORG.DBF
        : CL_OTHER.DBF
        : POLEST.DBF
        : AMMOEST.DBF

Used by: REINDX.PRG

ECHELON is an index tag

This index file is associated with database(s):
        : CSRCLIST.DBF
        : ECHELONS.DBF
        : MEDICAL.DBF
        : WATER.DBF

Used by: REINDX.PRG

LIN is an index tag

This index file is associated with database(s):
        : SRCLINQT.DBF
        : CSRCLINQ.DBF
        : LINLIST.DBF
        : AMMORATE.DBF
        : POLRATE.DBF

427

```
                          : POLEST.DBF
                          : AMMOEST.DBF

          Used by: REINDX.PRG


          ------------------------------------------------------------
          UNIQUELI is an index tag

          This index file is associated with database(s):
                          : SRCLINQT.DBF

          Used by: REINDX.PRG


          ------------------------------------------------------------
          DODIC is an index tag

          This index file is associated with database(s):
                          : AMMORATE.DBF
                          : DODIC.DBF
                          : AMMOEST.DBF

          Used by: REINDX.PRG


          ------------------------------------------------------------
          POL_TYPE is an index tag

          This index file is associated with database(s):
                          : POLRATE.DBF
                          : POLWGT.DBF
                          : POLEST.DBF

          Used by: REINDX.PRG


          ------------------------------------------------------------
          TASKORGNAM is an index tag

          This index file is associated with database(s):
                          : TASKORG.DBF
                          : ORDERTAS.DBF
                          : CL_OTHER.DBF
                          : POLEST.DBF
                          : AMMOEST.DBF

          Used by: REINDX.PRG


          ------------------------------------------------------------
          UNIQUEUN is an index tag

          This index file is associated with database(s):
                          : TASKORG.DBF

          Used by: REINDX.PRG


          ------------------------------------------------------------
          UNIQUETA is an index tag

          This index file is associated with database(s):
                          : TASKORG.DBF
                          : ORDERTAS.DBF
```

428

Used by: REINDX.PRG

NUMBER is an index tag

This index file is associated with database(s):
: ORDERTAS.DBF
: ORDERLST.DBF
: POLEST.DBF
: AMMOEST.DBF

Used by: REINDX.PRG

TASKPHAS is an index tag

This index file is associated with database(s):
: ORDERTAS.DBF

Used by: REINDX.PRG

PROFILENAM is an index tag

This index file is associated with database(s):
: ORDERTAS.DBF
: CPROFILE.DBF
: CL_OTHER.DBF

Used by: REINDX.PRG

TASKPHASE is an index tag

This index file is associated with database(s):
: ORDERTAS.DBF

Used by: REINDX.PRG

UNIQUENU is an index tag

This index file is associated with database(s):
: ORDERTAS.DBF

Used by: REINDX.PRG

PHASE is an index tag

This index file is associated with database(s):
: ORDERTAS.DBF

Used by: REINDX.PRG

UNIQUETN is an index tag

This index file is associated with database(s):
        : ORDERTAS.DBF

Used by: REINDX.PRG

PHASETAS is an index tag

This index file is associated with database(s):
        : ORDERTAS.DBF

Used by: REINDX.PRG

MED_NAME is an index tag

This index file is associated with database(s):
        : MEDICAL.DBF
        : CPROFILE.DBF

Used by: REINDX.PRG

WATER_NAME is an index tag

This index file is associated with database(s):
        : WATER.DBF
        : CPROFILE.DBF

Used by: REINDX.PRG

RATION is an index tag

This index file is associated with database(s):
        : CLASS_I.DBF

Used by: REINDX.PRG

PROFILE is an index tag

This index file is associated with database(s):
        : PROFILES.DBF
        : CPROFILE.DBF

Used by: REINDX.PRG

INTENSITY is an index tag

This index file is associated with database(s):
        : CPROFILE.DBF

Used by: REINDX.PRG

PROFILEDES is an index tag

This index file is associated with database(s):,
         : CPROFILE.DBF

Used by: REINDX.PRG

POSTURE is an index tag

This index file is associated with database(s):
         : CPROFILE.DBF

Used by: REINDX.PRG

CDE is an index tag

This index file is associated with database(s):
         : CPROFILE.DBF

Used by: REINDX.PRG

NAME is an index tag

This index file is associated with database(s):
         : ORDERLST.DBF

Used by: REINDX.PRG

DESCRIPTIO is an index tag

This index file is associated with database(s):
         : ORDERLST.DBF

Used by: REINDX.PRG

PARENT_UNI is an index tag

This index file is associated with database(s):
         : POLEST.DBF

Used by: REINDX.PRG

UNITDODI is an index tag

This index file is associated with database(s):
         : AMMOEST.DBF

Used by: REINDX.PRG

System: SURE 2.0
Author: MAJ John M. Friedson
04/16/94   12:51:19
Multiple Index File Parameter Summary

```
-----------------------------------------------------/-------------------------------------------------

24 multiple index files in the system
    D:\SURE\HELP.MDX
    D:\SURE\CSRCLIST.MDX
    D:\SURE\SRCLIST.MDX
    D:\SURE\SRCLINQT.MDX
    D:\SURE\CSRCLINQ.MDX
    D:\SURE\ECHELONS.MDX
    D:\SURE\LINLIST.MDX
    D:\SURE\AMMORATE.MDX
    D:\SURE\POLRATE.MDX
    D:\SURE\TASKORG.MDX
    D:\SURE\ORDERTAS.MDX
    D:\SURE\MEDICAL.MDX
    D:\SURE\WATER.MDX
    D:\SURE\CLASS_I.MDX
    D:\SURE\INTENSTY.MDX
    D:\SURE\PROFILES.MDX
    D:\SURE\CPROFILE.MDX
    D:\SURE\ORDERLST.MDX
    D:\SURE\CL_OTHER.MDX
    D:\SURE\POLWGT.MDX
    D:\SURE\DODIC.MDX
    D:\SURE\POLEST.MDX
    D:\SURE\AMMOEST.MDX
    D:\SURE\&FILEVAR
-----------------------------------------------------------------------------

HELP.MDX                Last updated:  04/10/94 at 19:25
    Tag: Program          (PROGRAM)
    Tag: Subject          (SUBJECT)

This multiple index file is associated with database(s):
        : HELP.DBF

Used by: STARTHLP.PRG
        : REINDX.PRG


-----------------------------------------------------------------------------
CSRCLIST.MDX            Last updated:  04/16/94 at 10:49
    Tag: Unit_desig       (UNIT_DESIG)
    Tag: Strength         (STRENGTH)
    Tag: Unitname         (UNITNAME)
    Tag: Src              (SRC)
    Tag: Echelon          (ECHELON)

This multiple index file is associated with database(s):
        : CSRCLIST.DBF

Used by: REINDX.PRG
        : BACKDBF.PRG
        : STARTCSR              (procedure in CREATESR.PRG)
        : STARTUNI              (procedure in EDITUNIT.PRG)
        : STARTED               (procedure in DELUNIT.PRG)
        : STARTASK              (procedure in CREATETA.PRG)
        : CALCPOL.PRG
        : CALCAMMO.PRG
```

```
                   : CALCOTHE.PRG
                   : STARTSRP              (procedure in SYSREP.PRG)
                   : STARTPLR              (procedure in PRINTPOL.PRG)


-------------------------------------------------------------------
SRCLIST.MDX           Last updated:  04/10/94 at 19:25
   Tag: Src              (SRC)
   Tag: Unit_desig       (UNIT_DESIG)

This multiple index file is associated with database(s):
          : SRCLIST.DBF

Used by: REINDX.PRG
          : STARTCSR              (procedure in CREATESR.PRG)
          : STARTUNI              (procedure in EDITUNIT.PRG)
          : STARTSRP              (procedure in SYSREP.PRG)


-------------------------------------------------------------------
SRCLINQT.MDX          Last updated:  04/10/94 at 19:26
   Tag: Lin              (LIN)
   Tag: Src              (SRC)
   Tag: Uniqueli         (LIN)

This multiple index file is associated with database(s):
          : SRCLINQT.DBF

Used by: REINDX.PRG
          : STARTCSR              (procedure in CREATESR.PRG)
          : STARTUNI              (procedure in EDITUNIT.PRG)
          : STARTSRP              (procedure in SYSREP.PRG)


-------------------------------------------------------------------
CSRCLINQ.MDX          Last updated:  04/16/94 at 10:49
   Tag: Lin              (LIN)
   Tag: Src              (SRC)
   Tag: Unitname         (UNITNAME)

This multiple index file is associated with database(s):
          : CSRCLINQ.DBF

Used by: REINDX.PRG
          : BACKDBF.PRG
          : STARTCSR              (procedure in CREATESR.PRG)
          : STARTUNI              (procedure in EDITUNIT.PRG)
          : STARTED               (procedure in DELUNIT.PRG)
          : CALCPOL.PRG
          : CALCAMMO.PRG
          : CALCOTHE.PRG
          : STARTSRP              (procedure in SYSREP.PRG)
          : STARTPLR              (procedure in PRINTPOL.PRG)


-------------------------------------------------------------------
ECHELONS.MDX          Last updated:  04/10/94 at 19:26
   Tag: Echelon          (ECHELON)

This multiple index file is associated with database(s):
          : ECHELONS.DBF

Used by: REINDX.PRG
```

```
        : ADDDATA                    (procedure in CREATESR.PRG)
        : STARTUNI                   (procedure in EDITUNIT.PRG)
        : CHANGEEH                   (procedure in EDITUNIT.PRG)
        : CALCOTHE.PRG


------------------------------------------------------------------
LINLIST.MDX              Last updated:  04/10/94 at 19:26
   Tag: Lin              (LIN)


This multiple index file is associated with database(s):
        : LINLIST.DBF


Used by: REINDX.PRG
        : STARTUNI                   (procedure in EDITUNIT.PRG)


------------------------------------------------------------------
AMMORATE.MDX            Last updated:  04/10/94 at 19:26
   Tag: Lin              (LIN)
   Tag: Dodic            (DODIC)


This multiple index file is associated with database(s):
        : AMMORATE.DBF


Used by: REINDX.PRG
        : STARTUNI                   (procedure in EDITUNIT.PRG)
        : CALCAMMO.PRG
        : STARTSRP                   (procedure in SYSREP.PRG)
        : STARTPLR                   (procedure in PRINTPOL.PRG)


------------------------------------------------------------------
POLRATE.MDX             Last updated:  04/10/94 at 19:26
   Tag: Lin              (LIN)
   Tag: Pol_type         (POL_TYPE)


This multiple index file is associated with database(s):
        : POLRATE.DBF


Used by: REINDX.PRG
        : STARTUNI                   (procedure in EDITUNIT.PRG)
        : CALCPOL.PRG
        : STARTSRP                   (procedure in SYSREP.PRG)
        : STARTPLR                   (procedure in PRINTPOL.PRG)


------------------------------------------------------------------
TASKORG.MDX             Last updated:  04/15/94 at 19:47
   Tag: Unitname         (UNITNAME)
   Tag: Uniqueun         (UNITNAME)
   Tag: Uniqueta         (TASKORGNAM)
   Tag: Taskorgnam       (TASKORGNAM)
   Tag: Src              (SRC)


This multiple index file is associated with database(s):
        : TASKORG.DBF


Used by: REINDX.PRG
        : BACKDBF.PRG
        : STARTED                    (procedure in DELUNIT.PRG)
        : STARTASK                   (procedure in CREATETA.PRG)
        : STARTOR                    (procedure in CREATEOR.PRG)
```

```
                : CALCPOL.PRG
                : CALCAMMO.PRG
                : CALCOTHE.PRG
                : STARTSRP            (procedure in SYSREP.PRG)
                : STARTPLR            (procedure in PRINTPOL.PRG)

---------------------------------------------------------------

ORDERTAS.MDX          Last updated:  04/15/94 at 19:46
  Tag: Uniqueta        (TASKORGNAM)
  Tag: Phase           (PHASE)
  Tag: Uniquetn        (TASKORGNAM+NUMBER)
  Tag: Number          (NUMBER)
  Tag: Taskorgnam      (TASKORGNAM)
  Tag: Taskphas        (TASKORGNAM+STR(PHASE))
  Tag: Profilenam      (PROFILENAM)
  Tag: Taskphase       (TASKORGNAM+STR(PHASE))
  Tag: Uniquenu        (NUMBER)
  Tag: Phasetas        (STR(PHASE)+TASKORGNAM)

This multiple index file is associated with database(s):
        : ORDERTAS.DBF

Used by: DELORDE.PRG
         : REINDX.PRG
         : BACKDBF.PRG
         : STARTED             (procedure in DELUNIT.PRG)
         : STARTOR             (procedure in CREATEOR.PRG)
         : CALCPOL.PRG
         : CALCAMMO.PRG
         : CALCOTHE.PRG
         : STARTSRP            (procedure in SYSREP.PRG)
         : STARTPLR            (procedure in PRINTPOL.PRG)

---------------------------------------------------------------

MEDICAL.MDX           Last updated:  04/10/94 at 19:26
  Tag: Med_name        (MED_NAME)
  Tag: Echelon         (ECHELON)

This multiple index file is associated with database(s):
        : MEDICAL.DBF

Used by: REINDX.PRG
         : STARTPRO            (procedure in CREATEPR.PRG)
         : STARTEPR            (procedure in EDITPROF.PRG)
         : CALCOTHE.PRG
         : STARTPLR            (procedure in PRINTPOL.PRG)

---------------------------------------------------------------

WATER.MDX             Last updated:  04/10/94 at 19:26
  Tag: Echelon         (ECHELON)
  Tag: Water_name      (WATER_NAME)

This multiple index file is associated with database(s):
        : WATER.DBF

Used by: REINDX.PRG
         : STARTPRO            (procedure in CREATEPR.PRG)
         : STARTEPR            (procedure in EDITPROF.PRG)
         : CALCOTHE.PRG
```

435

: STARTPLR   (procedure in PRINTPOL.PRG)

CLASS_I.MDX   Last updated: 04/10/94 at 19:26
   Tag: Ration   (RATION)

This multiple index file is associated with database(s):
         : CLASS_I.DBF

Used by: REINDX.PRG
         : STARTPRO   (procedure in CREATEPR.PRG)
         : STARTEPR   (procedure in EDITPROF.PRG)
         : STARTPLR   (procedure in PRINTPOL.PRG)

INTENSTY.MDX   Last updated: 03/30/94 at 7:34
   Tag: Profile   (PROFILE)
   Tag: Intensity   (INTENSITY)
   Tag: Profilenam   (PROFILENAM)
   Tag: Profiledes   (PROFILEDES)
   Tag: Posture   (POSTURE)
   Tag: Med_name   (MED_NAME)
   Tag: Water_name   (WATER_NAME)
   Tag: Cde   (CDE)

This multiple index file is associated with database(s):
         : INTENSTY.DBF

Used by: BACKDBF.PRG
         : STARTPRO   (procedure in CREATEPR.PRG)
         : STARTEPR   (procedure in EDITPROF.PRG)
         : CALCPOL.PRG
         : CALCAMMO.PRG
         : CALCOTHE.PRG
         : STARTPLR   (procedure in PRINTPOL.PRG)

PROFILES.MDX   Last updated: 04/10/94 at 19:26
   Tag: Profile   (PROFILE)

This multiple index file is associated with database(s):
         : PROFILES.DBF

Used by: REINDX.PRG
         : BACKDBF.PRG
         : STARTPRO   (procedure in CREATEPR.PRG)
         : STARTEPR   (procedure in EDITPROF.PRG)
         : CALCPOL.PRG
         : CALCAMMO.PRG
         : CALCOTHE.PRG
         : STARTPLR   (procedure in PRINTPOL.PRG)

CPROFILE.MDX   Last updated: 04/15/94 at 19:50
   Tag: Med_name   (MED_NAME)
   Tag: Water_name   (WATER_NAME)
   Tag: Profile   (PROFILE)
   Tag: Intensity   (INTENSITY)
   Tag: Profilenam   (PROFILENAM)

```
       Tag: Profiledes        (PROFILEDES)
       Tag: Posture           (POSTURE)
       Tag: Cde               (CDE)

This multiple index file is associated with database(s):
          : CPROFILE.DBF

Used by: REINDX.PRG
         : BACKDBF.PRG
         : STARTED              (procedure in DELUNIT.PRG)
         : STARTPRO             (procedure in CREATEPR.PRG)
         : STARTEPR             (procedure in EDITPROF.PRG)
         : STARTOR              (procedure in CREATEOR.PRG)
         : CALCPOL.PRG
         : CALCAMMO.PRG
         : CALCOTHE.PRG
         : STARTSRP             (procedure in SYSREP.PRG)
         : STARTPLR             (procedure in PRINTPOL.PRG)

----------------------------------------------------------------
ORDERLST.MDX          Last updated:  04/15/94 at 19:47
  Tag: Number            (NUMBER)
  Tag: Name              (NAME)
  Tag: Descriptio        (DESCRIPTIO)

This multiple index file is associated with database(s):
          : ORDERLST.DBF

Used by: DELORDE.PRG
         : REINDX.PRG
         : BACKDBF.PRG
         : STARTOR              (procedure in CREATEOR.PRG)
         : STARTORD             (procedure in ORDEREP1.PRG)
         : CALCPOL.PRG
         : CALCAMMO.PRG
         : CALCOTHE.PRG
         : STARTSRP             (procedure in SYSREP.PRG)
         : STARTPLR             (procedure in PRINTPOL.PRG)

----------------------------------------------------------------
CL_OTHER.MDX          Last updated:  04/12/94 at 21:59
  Tag: Unitname          (UNITNAME)
  Tag: Taskorgnam        (TASKORGNAM)
  Tag: Profilenam        (PROFILENAM)

This multiple index file is associated with database(s):
          : CL_OTHER.DBF

Used by: REINDX.PRG
         : STARTORD             (procedure in ORDEREP1.PRG)
         : CALCOTHE.PRG
         : STARTPLR             (procedure in PRINTPOL.PRG)

----------------------------------------------------------------
POLWGT.MDX            Last updated:  04/10/94 at 19:26
  Tag: Pol_type          (POL_TYPE)

This multiple index file is associated with database(s):
          : POLWGT.DBF
```

437

```
Used by: REINDX.PRG
       : CALCPOL.PRG
       : STARTPLR                    (procedure in PRINTPOL.PRG)


-----------------------------------------------------------------

DODIC.MDX              Last updated:  04/10/94 at 19:26
  Tag: Dodic              (DODIC)

This multiple index file is associated with database(s):
       : DODIC.DBF

Used by: REINDX.PRG
       : CALCAMMO.PRG
       : STARTPLR                    (procedure in PRINTPOL.PRG)


-----------------------------------------------------------------

POLEST.MDX             Last updated:  04/12/94 at 21:58
  Tag: Lin                (LIN)
  Tag: Pol_type           (POL_TYPE)
  Tag: Number             (NUMBER)
  Tag: Unitname           (UNITNAME)
  Tag: Taskorgnam         (TASKORGNAM)
  Tag: Parent_uni         (PARENT_UNI)

This multiple index file is associated with database(s):
       : POLEST.DBF

Used by: REINDX.PRG
       : CALCPOL.PRG
       : STARTPLR                    (procedure in PRINTPOL.PRG)


-----------------------------------------------------------------

AMMOEST.MDX            Last updated:  04/12/94 at 21:59
  Tag: Dodic              (DODIC)
  Tag: Lin                (LIN)
  Tag: Number             (NUMBER)
  Tag: Unitname           (UNITNAME)
  Tag: Taskorgnam         (TASKORGNAM)
  Tag: Unitdodi           (UNITNAME+LIN+DODIC)

This multiple index file is associated with database(s):
       : AMMOEST.DBF

Used by: REINDX.PRG
       : CALCAMMO.PRG
```

```
SURE.PRG
,     VARIABLE.MEM (memory file)
+----HELP.PRG
|    +----MEMOPAGR() (function  in MEMOPAGR.PRG)
+----SETCON.PRG
+----STARTHLP.PRG
|    |    HELP.DBF (database)
|    +----HELP.PRG...
+----MEMVARS.PRG
+----DEF_MENS    (procedure in SURE.PRG)
     +----SRC_PRO     (procedure in SURE.PRG)
     |    +----CREATESR.PRG
     |    |    +----STARTCSR     (procedure in CREATESR.PRG)
     |    |    |    |    CSRCLIST.DBF (database)
     |    |    |    |    TEMP.DBF (database)
     |    |    |    |    SRCLIST.DBF (database)
     |    |    |    |    SRCLINQT.DBF (database)
     |    |    |    |    CSRCLINQ.DBF (database)
     |    |    |    +----STARTHLP.PRG...
     |    |    +----CSRC_PRO     (procedure in CREATESR.PRG)
     |    |         +----COPYSRC     (procedure in CREATESR.PRG)
     |    |         |    +----SETBROWS.PRG
     |    |         |    +----ENDBROWS.PRG
     |    |         |    +----DISPUNIT     (procedure in CREATESR.PRG)
     |    |         +----SEEKSRC     (procedure in CREATESR.PRG)
     |    |         |    +----COPYSRC...     (procedure in CREATESR.PRG)
     |    |         +----CSRCENDA     (procedure in CREATESR.PRG)
     |    |         |    +----ADDDATA     (procedure in CREATESR.PRG)
     |    |         |    |    |    ECHELONS.DBF (database)
     |    |         |    |    +----ECHELON.PRG
     |    |         |    |    +----SETBROWS.PRG...
     |    |         |    |    +----ENDBROWS.PRG...
     |    |         |    |    +----DISPUNIT...     (procedure in CREATESR.PRG)
     |    |         |    +----COPYLINS     (procedure in CREATESR.PRG)
     |    |         |    |    |    TTEMP.DBF (database)
     |    |         |    |    |    TTTEMP.DBF (database)
     |    |         |    +----STARTCSR...     (procedure in CREATESR.PRG)
     |    |         +----COPYCSRC     (procedure in CREATESR.PRG)
     |    |         |    +----SETBROWS.PRG...
     |    |         |    +----ENDBROWS.PRG...
     |    |         |    +----DISPUNIT...     (procedure in CREATESR.PRG)
     |    |         +----VIEWUNIT     (procedure in CREATESR.PRG)
     |    |         |    +----SETBROWS.PRG...
     |    |         |    +----ENDBROWS.PRG...
     |    |         +----CSRCEND     (procedure in CREATESR.PRG)
     |    |              +----ADDDATA...     (procedure in CREATESR.PRG)
     |    |              +----COPYLINS...     (procedure in CREATESR.PRG)
     |    +----STARTHLP.PRG...
     |    +----EDITUNIT.PRG
     |    |    +----STARTUNI     (procedure in EDITUNIT.PRG)
     |    |    |    |    CSRCLIST.DBF (database)
     |    |    |    |    LINLIST.DBF (database)
     |    |    |    |    AMMORATE.DBF (database)
     |    |    |    |    POLRATE.DBF (database)
     |    |    |    |    SRCLIST.DBF (database)
     |    |    |    |    SRCLINQT.DBF (database)
     |    |    |    |    CSRCLINQ.DBF (database)
     |    |    |    |    ECHELONS.DBF (database)
     |    |    |    +----STARTHLP.PRG...
     |    |    |    +----HELP.PRG...
     |    |    |    +----ENDUNIT     (procedure in EDITUNIT.PRG)
     |    |    +----EDIU_PRO     (procedure in EDITUNIT.PRG)
     |    |         +----SELECTUNI     (procedure in EDITUNIT.PRG)
     |    |         |    +----SETBROWS.PRG...
     |    |         |    +----ENDBROWS.PRG...
     |    |         |    +----DISPUNIT...     (procedure in CREATESR.PRG)
     |    |         +----SELECTUN2     (procedure in EDITUNIT.PRG)
     |    |              +----SETBROWS.PRG...
```

439

```
            |           |     +----ENDBROWS.PRG...
            |           |     +----DISPUNIT...      (procedure in CREATESR.PRG)
            |           +----CHANGELI     (procedure in EDITUNIT.PRG)
            |           |     +----SETBROWS.PRG...
            |           |     +----ENDBROWS.PRG...
            |           |     +----DISPUNIT...      (procedure in CREATESR.PRG)
            |           +----ADDLINCA      (procedure in EDITUNIT.PRG)
            |           |     +----SETBROWS.PRG...
            |           |     +----ENDBROWS.PRG...
            |           |     +----DISPUNIT...      (procedure in CREATESR.PRG)
            |           +----DELLIN      (procedure in EDITUNIT.PRG)
            |           |     +----SETBROWS.PRG...
            |           |     +----ENDBROWS.PRG...
            |           |     +----DISPUNIT...      (procedure in CREATESR.PRG)
            |           +----EDITSTR     (procedure in EDITUNIT.PRG)
            |           |     +----DISPUNIT...      (procedure in CREATESR.PRG)
            |           +----CHANGEEH      (procedure in EDITUNIT.PRG)
            |           |     |    ECHELONS.DBF (database)
            |           |     +----SETBROWS.PRG...
            |           |     +----ENDBROWS.PRG...
            |           |     +----DISPUNIT...      (procedure in CREATESR.PRG)
            |           +----ENDUNIT...     (procedure in EDITUNIT.PRG)
            +----CUSTUNIT.PRG
            |     +----STARTUNI...      (procedure in EDITUNIT.PRG)
            |     +----CUST_PRO     (procedure in CUSTUNIT.PRG)
            |           +----UNITDATA      (procedure in CUSTUNIT.PRG)
            |           |     |    TEMP.DBF (database)
            |           |     +----ECHELON.PRG...
            |           |     +----DISPUNIT...      (procedure in CREATESR.PRG)
            |           +----CHANGELI...     (procedure in EDITUNIT.PRG)
            |           +----ADDLINCA...     (procedure in EDITUNIT.PRG)
            |           +----DELLIN...     (procedure in EDITUNIT.PRG)
            |           +----CHANGEEH...     (procedure in EDITUNIT.PRG)
            |           +----ENDUNIT...     (procedure in EDITUNIT.PRG)
            +----DELUNIT.PRG
            |     +----STARTED     (procedure in DELUNIT.PRG)
            |     |    |    CSRCLIST.DBF (database)
            |     |    |    CSRCLINQ.DBF (database)
            |     |    |    TASKORG.DBF (database)
            |     |    |    ORDERTAS.DBF (database)
            |     |    |    CPROFILE.DBF (database)
            |     |    +----STARTHLP.PRG...
            |     +----DELU_PRO     (procedure in DELUNIT.PRG)
            |           +----SELECTUNI...     (procedure in EDITUNIT.PRG)
            |           +----SELECTUN2...     (procedure in EDITUNIT.PRG)
            |           +----DELUAS     (procedure in DELUNIT.PRG)
            |           +----ENDDELU     (procedure in DELUNIT.PRG)
+----TASK_PRO     (procedure in SURE.PRG)
|     +----CREATETA.PRG
|     |     +----STARTASK     (procedure in CREATETA.PRG)
|     |     |    |    TASKORG.DBF (database)
|     |     |    |    CSRCLIST.DBF (database)
|     |     |    +----STARTHLP.PRG...
|     |     +----CRET_PRO     (procedure in CREATETA.PRG)
|     |           +----NAMETA      (procedure in CREATETA.PRG)
|     |           |     +----DISPTASK     (procedure in CREATETA.PRG)
|     |           +----ASSIGNUN     (procedure in CREATETA.PRG)
|     |           |    |    TEMP.DBF (database)
|     |           |    |    TTEMP.DBF (database)
|     |           |    +----DISPTASK...     (procedure in CREATETA.PRG)
|     |           +----ASSIGNU2     (procedure in CREATETA.PRG)
|     |           |    |    TEMP.DBF (database)
|     |           |    |    TTEMP.DBF (database)
|     |           |    +----DISPTASK...     (procedure in CREATETA.PRG)
|     |           +----UNITQTY     (procedure in CREATETA.PRG)
|     |           +----ENDTASKA     (procedure in CREATETA.PRG)
|     |           |         TEMP.DBF (database)
|     |           |         TTEMP.DBF (database)
|     |           +----ENDTASK     (procedure in CREATETA.PRG)
|     |                     TEMP.DBF (database)
|     |                     TTEMP.DBF (database)
|     +----STARTHLP.PRG...
|     +----CREATETC.PRG
```

440

```
    |     |    +----STARTASK...      (procedure in CREATETA.PRG)
    |     |    +----COPT_PRO    (procedure in CREATETC.PRG)
    |     |         +----FINDTA      (procedure in CREATETC.PRG)
    |     |         |    +----SETBROWS.PRG...
    |     |         |    +----ENDBROWS.PRG...
    |     |         |    +----DISPTASK...     (procedure in CREATETA.PRG)
    |     |         +----ASSIGNNA     (procedure in CREATETC.PRG)
    |     |         |    +----DISPTASK...     (procedure in CREATETA.PRG)
    |     |         +----ENDCOPT     (procedure in CREATETC.PRG)
    |     |         |         TEMP.DBF (database)
    |     |         +----ENDCOPTA     (procedure in CREATETC.PRG)
    |     |                   TEMP.DBF (database)
    |     +----EDITTASK.PRG
    |     |    +----STARTED...     (procedure in DELUNIT.PRG)
    |     |    +----FINDTA...     (procedure in CREATETC.PRG)
    |     |    +----EDIT_PRO     (procedure in EDITTASK.PRG)
    |     |         +----FINDTA...     (procedure in CREATETC.PRG)
    |     |         +----ASSIGNUN...     (procedure in CREATETA.PRG)
    |     |         +----ASSIGNU2...     (procedure in CREATETA.PRG)
    |     |         +----VIEWTASK     (procedure in EDITTASK.PRG)
    |     |         |    +----SETBROW2.PRG
    |     |         |    +----ENDBROWS.PRG...
    |     |         |    +----DISPTASK...     (procedure in CREATETA.PRG)
    |     |         +----VIEWTAS2     (procedure in EDITTASK.PRG)
    |     |         |    +----SETBROW2.PRG...
    |     |         |    +----ENDBROWS.PRG...
    |     |         |    +----DISPTASK...     (procedure in CREATETA.PRG)
    |     |         +----UNASSIGU     (procedure in EDITTASK.PRG)
    |     |         |    +----DISPTASK...     (procedure in CREATETA.PRG)
    |     |         +----UNASSIG2     (procedure in EDITTASK.PRG)
    |     |         |    +----DISPTASK...     (procedure in CREATETA.PRG)
    |     |         +----ENDEDIT     (procedure in EDITTASK.PRG)
    |     +----DELTASK.PRG
    |          +----STARTED...     (procedure in DELUNIT.PRG)
    |          +----DELT_PRO     (procedure in DELTASK.PRG)
    |               +----FINDTA...     (procedure in CREATETC.PRG)
    |               +----DELTAS     (procedure in DELTASK.PRG)
    |               +----ENDDELT     (procedure in DELTASK.PRG)
+----PROF_PRO     (procedure in SURE.PRG)
|    +----CREATEPR.PRG
|    |    +----STARTPRO     (procedure in CREATEPR.PRG)
|    |    |    |    MEDICAL.DBF (database)
|    |    |    |    WATER.DBF (database)
|    |    |    |    CLASS_I.DBF (database)
|    |    |    |    POSTURE.DBF (database)
|    |    |    |    INTENSTY.DBF (database)
|    |    |    |    PROFILES.DBF (database)
|    |    |    |    CPROFILE.DBF (database)
|    |    |    +----STARTHLP.PRG...
|    |    +----CREP_PRO     (procedure in CREATEPR.PRG)
|    |         +----NEWPROFI     (procedure in CREATEPR.PRG)
|    |         |    |    TEMP.DBF (database)
|    |         |    +----SETBROWS.PRG...
|    |         |    +----ENDBROWS.PRG...
|    |         |    +----DISPPROF     (procedure in CREATEPR.PRG)
|    |         +----VIEWPROF     (procedure in CREATEPR.PRG)
|    |         |    +----SETBROWS.PRG...
|    |         |    +----ENDBROWS.PRG...
|    |         |    +----DISPPROF...     (procedure in CREATEPR.PRG)
|    |         +----ENDCREP     (procedure in CREATEPR.PRG)
|    +----STARTHLP.PRG...
|    +----EDITPROF.PRG
|    |    +----STARTEPR     (procedure in EDITPROF.PRG)
|    |    |    |    MEDICAL.DBF (database)
|    |    |    |    WATER.DBF (database)
|    |    |    |    CLASS_I.DBF (database)
|    |    |    |    INTENSTY.DBF (database)
|    |    |    |    PROFILES.DBF (database)
|    |    |    |    POSTURE.DBF (database)
|    |    |    |    CPROFILE.DBF (database)
|    |    |    +----STARTHLP.PRG...
|    |    +----EDIP_PRO     (procedure in EDITPROF.PRG)
|    |         +----SELEPROF     (procedure in EDITPROF.PRG)
```

441

```
|    |               |    +----SETBROWS.PRG...
|    |               |    +----ENDBROWS.PRG...
|    |               |    +----DISPPROF...      (procedure in CREATEPR.PRG)
|    |            +----CHANGEAP     (procedure in EDITPROF.PRG)
|    |            |    +----SETBROWS.PRG...
|    |            |    +----ENDBROWS.PRG...
|    |            |    +----DISPPROF...      (procedure in CREATEPR.PRG)
|    |            +----CHANGEAC     (procedure in EDITPROF.PRG)
|    |            |    +----SETBROWS.PRG...
|    |            |    +----ENDBROWS.PRG...
|    |            |    +----DISPPROF...        (procedure in CREATEPR.PRG)
|    |            +----CHANGEI      (procedure in EDITPROF.PRG)
|    |            |    +----SETBROWS.PRG...
|    |            |    +----ENDBROWS.PRG...
|    |            |    +----DISPPROF...        (procedure in CREATEPR.PRG)
|    |            +----CHANGEIF     (procedure in EDITPROF.PRG)
|    |            |    +----DISPPROF...        (procedure in CREATEPR.PRG)
|    |            +----CHANGECI     (procedure in EDITPROF.PRG)
|    |            |    +----SETBROWS.PRG...
|    |            |    +----ENDBROWS.PRG...
|    |            |    +----DISPPROF...        (procedure in CREATEPR.PRG)
|    |            +----CHANGEH2     (procedure in EDITPROF.PRG)
|    |            |    +----SETBROWS.PRG...
|    |            |    +----ENDBROWS.PRG...
|    |            |    +----DISPPROF...        (procedure in CREATEPR.PRG)
|    |            +----CHANGEME     (procedure in EDITPROF.PRG)
|    |            |    +----SETBROWS.PRG...
|    |            |    +----ENDBROWS.PRG...
|    |            |    +----DISPPROF...        (procedure in CREATEPR.PRG)
|    |            +----ENDCREP...      (procedure in CREATEPR.PRG)
|    +----DELPROF.PRG
|    |    +----STARTED...      (procedure in DELUNIT.PRG)
|    |    +----DELP_PRO     (procedure in DELPROF.PRG)
|    |         +----SELECTPRO       (procedure in DELPROF.PRG)
|    |         |    +----SETBROWS.PRG...
|    |         |    +----ENDBROWS.PRG...
|    |         |    +----DISPPROF...       (procedure in CREATEPR.PRG)
|    |         +----DELPRO    (procedure in DELPROF.PRG)
|    |         +----ENDDELP     (procedure in DELPROF.PRG)
|    +----CRECUSPR.PRG
|         +----STARTPRO...      (procedure in CREATEPR.PRG)
|         +----CRCP_PRO     (procedure in CRECUSPR.PRG)
|              +----NEWCUSPR     (procedure in CRECUSPR.PRG)
|              |    |    TEMP.DBF (database)
|              |    +----SETBROWS.PRG...
|              |    +----ENDBROWS.PRG...
|              +----VIEWCUSP     (procedure in CRECUSPR.PRG)
|              |    +----SETBROWS.PRG...
|              |    +----ENDBROWS.PRG...
|              |    +----DISPPRO     (procedure in CRECUSPR.PRG)
|              +----EDITCUSP     (procedure in CRECUSPR.PRG)
|              |    +----SETBROWS.PRG...
|              |    +----ENDBROWS.PRG...
|              +----ENDCUSPR     (procedure in CRECUSPR.PRG)
|              +----DELCUSPR     (procedure in CRECUSPR.PRG)
|                   +----SETBROWS.PRG...
|                   +----ENDBROWS.PRG...
|                   +----DISPPRO...     (procedure in CRECUSPR.PRG)
+----ORDE_PRO     (procedure in SURE.PRG)
|    +----CREATEOR.PRG
|    |    +----STARTOR       (procedure in CREATEOR.PRG)
|    |    |    |    CPROFILE.DBF (database)
|    |    |    |    TASKORG.DBF (database)
|    |    |    |    ORDERTAS.DBF (database)
|    |    |    |    ORDERLST.DBF (database)
|    |    |    +----STARTHLP.PRG...
|    |    +----TOMP_PRO     (procedure in CREATEOR.PRG)
|    |    |    +----1PROALL       (procedure in CREATEOR.PRG)
|    |    |    |    +----FINDPRO      (procedure in CREATEOR.PRG)
|    |    |    |         +----SETBROWS.PRG...
|    |    |    |         +----ENDBROWS.PRG...
|    |    |    |         +----DISPPROF...        (procedure in CREATEPR.PRG)
|    |    |    +----1PROONE       (procedure in CREATEOR.PRG)
```

442

```
|    |    |    |    +----SETBROWS.PRG...
|    |    |    |    +----ENDBROWS.PRG...
|    |    |    |    +----FINDPRO...       (procedure in CREATEOR.PRG)
|    |    |    +----1PRO1PRI    (procedure in CREATEOR.PRG)
|    |    |    |    +----FINDPRO...       (procedure in CREATEOR.PRG)
|    |    |    +----PHASEBY    (procedure in CREATEOR.PRG)
|    |    |    |    +----VIEWTOP      (procedure in CREATEOR.PRG)
|    |    |    |    |    +----SETBROWS.PRG...
|    |    |    |    |    +----ENDBROWS.PRG...
|    |    |    |    |    +----FINDPRO...       (procedure in CREATEOR.PRG)
|    |    |    +----VIEWTOP...      (procedure in CREATEOR.PRG)
|    |    +----ORDS_PRO    (procedure in CREATEOR.PRG)
|    |    |    +----STARTOR...      (procedure in CREATEOR.PRG)
|    |    |    +----CREATEO    (procedure in CREATEOR.PRG)
|    |    |    |    |    TEMP.DBF (database)
|    |    |    |    +----DISPORD     (procedure in CREATEOR.PRG)
|    |    |    +----VIEWORDE      (procedure in CREATEOR.PRG)
|    |    |    |    +----SETBROWS.PRG...
|    |    |    |    +----ENDBROWS.PRG...
|    |    |    |    +----DISPORD2      (procedure in CREATEOR.PRG)
|    |    |    |    +----DISPORD3     (procedure in ORDEREP1.PRG)
|    |    |    +----ASSIGNTA      (procedure in CREATEOR.PRG)
|    |    |    |         TTEMP.DBF (database)
|    |    |    |         TEMP.DBF (database)
|    |    |    +----ENDORDR      (procedure in CREATEOR.PRG)
|    +----STARTHLP.PRG...
|    +----EDITORDE.PRG
|    |    +----STARTOR...      (procedure in CREATEOR.PRG)
|    |    +----TOMP_PRO...      (procedure in CREATEOR.PRG)
|    |    +----STRE_PRO    (procedure in EDITORDE.PRG)
|    |    |    +----1STRALL     (procedure in EDITORDE.PRG)
|    |    |    +----1STRONE     (procedure in EDITORDE.PRG)
|    |    |    |    +----SETBROWS.PRG...
|    |    |    |    +----ENDBROWS.PRG...
|    |    |    +----1STR1PRI     (procedure in EDITORDE.PRG)
|    |    |    +----PHASEBYS     (procedure in EDITORDE.PRG)
|    |    |    |    +----SETBROW2.PRG...
|    |    |    |    +----ENDBROWS.PRG...
|    |    |    +----VIEWSTR     (procedure in EDITORDE.PRG)
|    |    |    |    +----SETBROWS.PRG...
|    |    |    |    +----ENDBROWS.PRG...
|    |    +----ORDT_PRO     (procedure in EDITORDE.PRG)
|    |    |    +----STARTOR...      (procedure in CREATEOR.PRG)
|    |    |    +----GETORDER     (procedure in EDITORDE.PRG)
|    |    |    |    +----SETBROWS.PRG...
|    |    |    |    +----ENDBROWS.PRG...
|    |    |    |    +----DISPORD2...      (procedure in CREATEOR.PRG)
|    |    |    |    +----VIEWORDE...      (procedure in CREATEOR.PRG)
|    |    |    |    +----DISPORD...      (procedure in CREATEOR.PRG)
|    |    |    |    +----DISPORD3...      (procedure in ORDEREP1.PRG)
|    |    |    |    +----CALCPOL.PRG
|    |    |    |    |    |    TASKORG.DBF (database)
|    |    |    |    |    |    CSRCLIST.DBF (database)
|    |    |    |    |    |    ORDERLST.DBF (database)
|    |    |    |    |    |    CPROFILE.DBF (database)
|    |    |    |    |    |    PROFILES.DBF (database)
|    |    |    |    |    |    POLRATE.DBF (database)
|    |    |    |    |    |    POSTURE.DBF (database)
|    |    |    |    |    |    INTENSTY.DBF (database)
|    |    |    |    |    |    POLWGT.DBF (database)
|    |    |    |    |    |    CSRCLINQ.DBF (database)
|    |    |    |    |    |    ORDERTAS.DBF (database)
|    |    |    |    |    |    POLEST.DBF (database)
|    |    |    |    |    +----STARTHLP.PRG...
|    |    |    |    |    +----POLCALC      (procedure in PRINTPOL.PRG)
|    |    |    |    +----CALCAMMO.PRG
|    |    |    |    |    |    TASKORG.DBF (database)
|    |    |    |    |    |    CSRCLIST.DBF (database)
|    |    |    |    |    |    ORDERLST.DBF (database)
|    |    |    |    |    |    CPROFILE.DBF (database)
|    |    |    |    |    |    PROFILES.DBF (database)
|    |    |    |    |    |    AMMORATE.DBF (database)
|    |    |    |    |    |    POSTURE.DBF (database)
```

443

```
      |        |         :      |    :    INTENSTY.DBF (database)
      |        |         :      |    :    CSRCLINQ.DBF (database)
      |        |         :      |    :    ORDERTAS.DBF (database)
      |        |         |      |    |    DODIC.DBF (database)
      |        |         |      |    |    AMMOEST.DBF (database)
      |        |         |      :    +----STARTHLP.PRG...
      |        |         |      :    +----AMMOCALC     (procedure in PRINTAMM.PRG)
      |        |         |      +----CALCOTHE.PRG
      |        |         :      |         TASKORG.DBF (database)
      |        |         :      |         CSRCLIST.DBF (database)
      |        |         |      |         ORDERLST.DBF (database)
      |        |         :      |         CPROFILE.DBF (database)
      |        |         :      |         PROFILES.DBF (database)
      |        |         |      |         POSTURE.DBF (database)
      |        |         |      |         INTENSTY.DBF (database)
      |        |         |      |         CSRCLINQ.DBF (database)
      |        |         |      |         ORDERTAS.DBF (database)
      |        |         |      |         MEDICAL.DBF (database)
      |        |         |      |         WATER.DBF (database)
      |        |         |      |         ECHELONS.DBF (database)
      |        |         |      |         CL_OTHER.DBF (database)
      |        |         |      +----STARTHLP.PRG...
      |        |         |      +----OTHERCAL     (procedure in CALCOTHE.PRG)
      |        |    +----VIEWORDE...     (procedure in CREATEOR.PRG)
      |        |    +----DISPORD2...     (procedure in CREATEOR.PRG)
      |        |    +----ASSIGNTA...     (procedure in CREATEOR.PRG)
      |        |    +----EDITPLEN     (procedure in EDITORDE.PRG)
      |        |    +----ENDORDR...     (procedure in CREATEOR.PRG)
      |    +----DELORDE.PRG
      |         |    ORDERTAS.DBF (database)
      |         |    ORDERLST.DBF (database)
      |         +----STARTHLP.PRG...
      |         +----DORD_PRO     (procedure in DELORDE.PRG)
      |              +----GETORDER...     (procedure in EDITORDE.PRG)
      +----REPO_PRO     (procedure in SURE.PRG)
      |    +----ORDEREP1.PRG
      |    |    +----ORDR_PRO     (procedure in ORDEREP1.PRG)
      |    |         +----STARTORD     (procedure in ORDEREP1.PRG)
      |    |         |         CL_OTHER.DBF (database)
      |    |         |         ORDERLST.DBF (database)
      |    |         +----OLDORD     (procedure in ORDEREP1.PRG)
      |    |         +----GETORDER...     (procedure in EDITORDE.PRG)
      |    |         +----DISPORD3...     (procedure in ORDEREP1.PRG)
      |    |         +----PRINTPOL.PRG
      |    |         |    +----STARTPLR     (procedure in PRINTPOL.PRG)
      |    |         |    |    |    TASKORG.DBF (database)
      |    |         |    |    |    CSRCLIST.DBF (database)
      |    |         |    |    |    ORDERLST.DBF (database)
      |    |         |    |    |    CPROFILE.DBF (database)
      |    |         |    |    |    PROFILES.DBF (database)
      |    |         |    |    |    POLRATE.DBF (database)
      |    |         |    |    |    POSTURE.DBF (database)
      |    |         |    |    |    INTENSTY.DBF (database)
      |    |         |    |    |    POLWGT.DBF (database)
      |    |         |    |    |    CSRCLINQ.DBF (database)
      |    |         |    |    |    ORDERTAS.DBF (database)
      |    |         |    |    |    DODIC.DBF (database)
      |    |         |    |    |    POLEST.DBF (database)
      |    |         |    |    |    AMMOEST.DBF (database)
      |    |         |    |    |    AMMORATE.DBF (database)
      |    |         |    |    |    MEDICAL.DBF (database)
      |    |         |    |    |    WATER.DBF (database)
      |    |         |    |    |    CLASS_I.DBF (database)
      |    |         |    |    |    CL_OTHER.DBF (database)
      |    |         |    |    +----STARTHLP.PRG...
      |    |         |    +----DETA_PRO     (procedure in PRINTPOL.PRG)
      |    |         |    +----PRIN_PRO     (procedure in SYSREP.PRG)
      |    |         |         +----PRNSETUP.PRG
      |    |         |         |         VARIABLE.MEM (memory file)
      |    |         |         +----STARTPLR...     (procedure in PRINTPOL.PRG)
      |    |         |         +----REPORTHD     (procedure in PRINTPOL.PRG)
      |    |         |         +----TFHEAD     (procedure in PRINTPOL.PRG)
      |    |         |         +----LINPOL     (procedure in PRINTPOL.PRG)
```

444

```
|  . |            |         +----POLCALC...     (procedure in PRINTPOL.PRG)
|  . |            |         +----PRINTPOL     (procedure in PRINTPOL.PRG)
|    |            .         +----PTASKPOL     (procedure in PRINTPOL.PRG)
|    |            |         +----PORDEPOL     (procedure in PRINTPOL.PRG)
|    |            |         +----LINAMM     (procedure in PRINTAMM.PRG)
|    |            |         +----AMMOCALC...     (procedure in PRINTAMM.PRG)
|    |            |         +----PUNITAMM     (procedure in PRINTAMM.PRG)
|    |            |         +----PTASKAMM     (procedure in PRINTAMM.PRG)
|    |            |         +----PORDEAMM     (procedure in PRINTAMM.PRG)
|    |            |         +----OTHECALC     (procedure in PRINTOTH.PRG)
|    |            |         +----PUNITOTH     (procedure in PRINTOTH.PRG)
|    |            |         +----PTASKOTH     (procedure in PRINTOTH.PRG)
|    |            |         +----PORDEOTH     (procedure in PRINTOTH.PRG)
|    |            +----PRINTAMM.PRG
|    |            |    +----STARTPLR...     (procedure in PRINTPOL.PRG)
|    |            |    +----DETA_PRO...     (procedure in PRINTPOL.PRG)
|    |            |    +----PRIN_PRO...     (procedure in SYSREP.PRG)
|    |            +----PRINTOTH.PRG
|    |            |    +----DETA_PRO...     (procedure in PRINTPOL.PRG)
|    |            |    +----STARTPLR...     (procedure in PRINTPOL.PRG)
|    |            |    +----PRIN_PRO...     (procedure in SYSREP.PRG)
|    |            +----ORDREND     (procedure in ORDEREP1.PRG)
|    +----STARTHLP.PRG...
|    +----SYSREP.PRG
|         +----STARTSRP     (procedure in SYSREP.PRG)
|         |    |    TASKORG.DBF (database)
|         |    |    CSRCLIST.DBF (database)
|         |    |    ORDERLST.DBF (database)
|         |    |    CPROFILE.DBF (database)
|         |    |    SRCLINQT.DBF (database)
|         |    |    SRCLIST.DBF (database)
|         |    |    AMMORATE.DBF (database)
|         |    |    POLRATE.DBF (database)
|         |    |    CSRCLINQ.DBF (database)
|         |    |    ORDERTAS.DBF (database)
|         |    +----STARTHLP.PRG...
|         +----PRIN_PRO...     (procedure in SYSREP.PRG)
|         +----SYSR_PRO     (procedure in SYSREP.PRG)
|              +----SRCRPT     (procedure in SYSREP.PRG)
|              |    +----SRCLST     (procedure in SYSREP.PRG)
|              +----SRCLINR     (procedure in SYSREP.PRG)
|              |    +----GETSRC     (procedure in SYSREP.PRG)
|              |    |    +----SETBROWS.PRG...
|              |    |    +----ENDBROWS.PRG...
|              |    |    +----DISPSRC     (procedure in SYSREP.PRG)
|              |    +----LINLST     (procedure in SYSREP.PRG)
|              +----UNITRPT     (procedure in SYSREP.PRG)
|              |    +----UNITLST     (procedure in SYSREP.PRG)
|              +----UNITLINR     (procedure in SYSREP.PRG)
|              |    +----UNITLIN     (procedure in SYSREP.PRG)
|              +----PROFILER     (procedure in SYSREP.PRG)
|              |    +----PROFLST     (procedure in SYSREP.PRG)
|              +----TASKORGR     (procedure in SYSREP.PRG)
|              |    +----TASKLST     (procedure in SYSREP.PRG)
|              +----TASKUNIT     (procedure in SYSREP.PRG)
|              |    +----TASKUNI     (procedure in SYSREP.PRG)
|              +----ORDERRPT     (procedure in SYSREP.PRG)
|              |    +----ORDERLST     (procedure in SYSREP.PRG)
|              +----ORDERTAS     (procedure in SYSREP.PRG)
|              |    +----ORDERTP     (procedure in SYSREP.PRG)
|              +----ORDTASUN     (procedure in SYSREP.PRG)
|              |    +----ORDERTUP     (procedure in SYSREP.PRG)
|              +----PRNSETUP.PRG...
|              +----SYSREND     (procedure in SYSREP.PRG)
+----UTIL_PRO     (procedure in SURE.PRG)
|    +----REINDX.PRG
|    |         HELP.DBF (database)
|    |         CSRCLIST.DBF (database)
|    |         SRCLIST.DBF (database)
|    |         SRCLINQT.DBF (database)
|    |         CSRCLINQ.DBF (database)
|    |         ECHELONS.DBF (database)
|    |         LINLIST.DBF (database)
```

445

```
|     |              AMMORATE.DBF (database)
|     |              POLRATE.DBF (database)
|     |              TASKORG.DBF (database)
|     |              ORDERTAS.DBF (database)
|     |              MEDICAL.DBF (database)
|     |              WATER.DBF (database)
|     |              CLASS_I.DBF (database)
|     |              PROFILES.DBF (database)
|     |              CPROFILE.DBF (database)
|     |              ORDERLST.DBF (database)
|     |              POLWGT.DBF (database)
|     |              DODIC.DBF (database)
|     |              POLEST.DBF (database)
|     |              AMMOEST.DBF (database)
|     |              CL_OTHER.DBF (database)
|     |              PROGRAM (index file)
|     |              SUBJECT (index file)
|     |              SRC (index file)
|     |              UNIT_DESIG (index file)
|     |              STRENGTH (index file)
|     |              UNITNAME (index file)
|     |              ECHELON (index file)
|     |              LIN (index file)
|     |              UNIQUELI (index file)
|     |              DODIC (index file)
|     |              POL_TYPE (index file)
|     |              TASKORGNAM (index file)
|     |              UNIQUEUN (index file)
|     |              UNIQUETA (index file)
|     |              NUMBER (index file)
|     |              TASKPHAS (index file)
|     |              PROFILENAM (index file)
|     |              TASKPHASE (index file)
|     |              UNIQUENU (index file)
|     |              PHASE (index file)
|     |              UNIQUETN (index file)
|     |              PHASETAS (index file)
|     |              MED_NAME (index file)
|     |              WATER_NAME (index file)
|     |              RATION (index file)
|     |              PROFILE (index file)
|     |              INTENSITY (index file)
|     |              PROFILEDES (index file)
|     |              POSTURE (index file)
|     |              CDE (index file)
|     |              NAME (index file)
|     |              DESCRIPTIO (index file)
|     |              PARENT_UNI (index file)
|     |              UNITDODI (index file)
|     +----STARTHLP.PRG...
|     +----BACKDBF.PRG
|     |              CSRCLIST.DBF (database)
|     |              &FILEVAR.DBF (database)
|     |              CSRCLINQ.DBF (database)
|     |              TASKORG.DBF (database)
|     |              ORDERTAS.DBF (database)
|     |              PROFILES.DBF (database)
|     |              INTENSTY.DBF (database)
|     |              CPROFILE.DBF (database)
|     |              ORDERLST.DBF (database)
|     +----RESTOREF.PRG
|     |     +----RESTORE       (procedure in RESTOREF.PRG)
|     |     |        &FILEVAR.DBF (database)
|     |     |        &FILEVAR2.DBF (database)
|     |     +----GETSTATU      (procedure in RESTOREF.PRG)
|     +----PRNSETUP.PRG...
|     +----COMPRESS.PRG
|     |              VARIABLE.MEM (memory file)
|     +----UNCOMPRE.PRG
+----HELP_PRO     (procedure in SURE.PRG)
|     +----HELPNDX.PRG
|     |     +----SETBROWS.PRG...
|     |     +----ENDBROWS.PRG...
```

```
|     |     +----MEMOPAGR() ... (function  in MEMOPAGR.PRG)
|     +----STARTHLP.PRG...
|     +----PRNTHELP.PRG
|     |     +----PHEL_PRO      (procedure in PRNTHELP.PRG)
|     |            +----PRNONEH      (procedure in PRNTHELP.PRG)
|     |            |     +----BROWHELP     (procedure in PRNTHELP.PRG)
|     |            |     |     +----SETBROWS.PRG...
|     |            |     |     +----ENDBROWS.PRG...
|     |            |     +----PRNTMEMO.PRG
|     |            +----PRNALLH      (procedure in PRNTHELP.PRG)
|     |            |     +----PRNTMEMO.PRG...
|     |            +----ENDPHEL      (procedure in PRNTHELP.PRG)
|     +----EDITHELP.PRG
|            +----SETBROWS.PRG...
|            +----ENDBROWS.PRG...
|            +----MEMOPAGR() ... (function  in MEMOPAGR.PRG)
+----QUIT_PRO      (procedure in SURE.PRG)
       +----CLOSEALL.PRG
 : STARTPLR                  (procedure in PRINTSOL.PRG)
```

447

BIBLIOGRAPHY

Books

_____dBase IV v2.0 Language Reference. Scotts Valley, CA: Borland International, Inc., 1993.

Simpson, Alan, dBase III Plus Programmer's Reference Guide. San Francisco, CA: Sybex Inc., 1987.

Manuals

U.S. Army, FM 101-10-1/2 Staff Officers Field Manual-Organizational, Technical, and Logistical Data, Planning Factors (Volumes 1 & 2). Washington: Department of the Army, 7 October 1987.

Memorandums and Documents

U.S. Army, "Agreement between Commander, CASCOM and Director, SAMS for the production of SURE 2.0." Fort Leavenworth, KS and Fort Lee, VA, 24 January 1994.

U.S. Army, "CASCOM SURE 2.0 Specifications." Fort Lee: VA, not dated.

Computer Software

_____dBase IV Version 1.5. Scotts Valley, CA: Borland International, Inc., 1992.

_____dBase Compiler for DOS. Scotts Valley, CA: Borland International, Inc., 1993.

Kennamer, Walter J. SNAP! Version 5.0 Documentation and Development System for FoxPro, dBASE, Clipper, and Other X-Base Systems. Toledo, OH: 1991.

_____Lotus 1-2-3 Release 2.2. Cambridge, MA: Lotus Development Corporation, 1989.

Mead, CPT Lawrence, and Kimbrell, Dorsey. Supply Usage Requirements Estimator. Fort Lee, VA: March 1994.